



André Marques Mano Coelho Monteiro

Licenciado em Engenharia Mecânica

Controlo e Estudo Dinâmico de Ascensores para Otimização do Conforto Humano

Dissertação para obtenção do Grau de
Mestre em Engenharia Mecânica

Orientador: Jorge Joaquim Pamies Teixeira, Prof. Catedrático,
FCT-UNL

Co-orientador: Luís Filipe Figueira de Brito Palma, Prof. Auxiliar,
FCT-UNL

Júri:

Presidente: Carla Maria Moreira Machado

Arguente: João Almeida das Rosas

Vogal: Jorge Joaquim Pamies Teixeira



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Março, 2016





André Marques Mano Coelho Monteiro

Licenciado em Engenharia Mecânica

Controlo e Estudo Dinâmico de Ascensores para Otimização do Conforto Humano

Dissertação para obtenção do Grau de
Mestre em Engenharia Mecânica

Orientador: Jorge Joaquim Pamies Teixeira, Prof. Catedrático,
FCT-UNL

Co-orientador: Luís Filipe Figueira de Brito Palma, Prof. Auxiliar,
FCT-UNL

Júri:

Presidente: Carla Maria Moreira Machado

Arguente: João Almeida das Rosas

Vogal: Jorge Joaquim Pamies Teixeira

Março, 2016



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Controlo e Estudo Dinâmico de Ascensores para Otimização do Conforto Humano

Copyright © 2016 André Marques Mano Coelho Monteiro, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Agradecimentos

Registo o meu profundo reconhecimento para com o meu orientador, Professor Jorge Pamies Teixeira, por me ter acolhido como mestrando. Expresso a minha gratidão para com o Professor Luís Brito Palma, na qualidade de co-orientador, por ter contribuído para resolução de problemas que foram surgindo ao longo deste estudo, nas áreas de Automação e Controlo.

Uma palavra de apreço à Faculdade de Ciências e Tecnologia pela confiança depositada e pelas condições oferecidas para a concretização do estudo que agora se apresenta.

Agradeço aos meus colegas de laboratório André Quintanova, Liliana Sequeira e Vasco Brito pelo apoio, sugestões e esclarecimento de pequenas dúvidas.

Grato ao Lino Estevão por me ensinar formas de estruturar o código no Arduino, à Sónia Correia pela motivação que me deu para iniciar a escrita desta dissertação e ao Pedro Riscado, confidente e amigo, com quem partilhei ideias.

Por fim agradeço aos meus pais e irmã que me apoiaram nos momentos mais difíceis, e aos meus amigos Carlos, David, Diogo, Gonçalo, Kharallos, Miranda, Pires e Tabby que me acompanharam ao longo deste processo.

Resumo

Nesta dissertação foram estudadas formas de proporcionar conforto ergonómico aos passageiros de um ascensor. O estudo foi realizado no protótipo de elevador monta-cargas desenvolvido no âmbito de uma tese de mestrado. O elevador encontra-se no laboratório 2.3 do Departamento de Engenharia Eletrotécnica da FCT-UNL.

O objetivo central do estudo foi o desenvolvimento de um sistema de baixo custo que avalia o conforto ergonómico em ascensores. Com base na norma ISO 18738 foi definida uma escala de conforto cuja aplicabilidade foi verificada através de um inquérito, no qual os inquiridos pontuavam de 1 a 5 o conforto sentido numa viagem de ascensor. O sistema tem também a funcionalidade de estimar os perfis de deslocamento, velocidade e aceleração da cabina de um ascensor.

Desenvolveu-se um algoritmo de controlo que recorre ao referido sistema para monitorizar parâmetros de conforto e proporcionar uma viagem confortável. O conforto foi garantido pelo controlo da velocidade da máquina de tração.

O sistema desenvolvido mostrou ser possível avaliar e melhorar o desempenho ergonómico nos ascensores.

Palavras-chave:

Sistema de baixo custo, conforto ergonómico, modelo de elevador monta-cargas, ascensor, controlo de velocidade.

Abstract

In this thesis it was studied how to provide ergonomic comfort on elevators. The study was conducted on a model of a cargo elevator developed for a master's thesis. The model is at 2.3 laboratory of the Department of Electrical Engineer of FCT-UNL.

The main objective of this study was to develop a low-cost System that evaluated the ergonomic comfort of a lift ride. Based on the ISO 18738 standard was defined a comfort scale whose applicability was verified through a survey, in which respondents punctuated from 1 to 5 the comfort of a lift ride. The System also can estimate the displacement, speed and acceleration profiles of the elevator cabin.

It was developed a control algorithm that uses the System for monitoring comfort parameters and provide a comfortable ride. The ergonomic comfort was achieved by controlling the motor's speed.

The developed system proved to be possible to assess and improve the ergonomic comfort of elevators.

Keywords: Low-cost System, ergonomic ride, model of a cargo elevator, elevator, speed control.

Índice

Índice	xi
Lista de Figuras	xv
Lista de Tabelas	xix
Lista de Siglas	xxi
1 Introdução	1
1.1 Contexto e Motivação	1
1.2 Objetivos	2
1.3 Contribuições	2
1.4 Estrutura da Dissertação	3
2 Revisão do Estado da Arte	5
2.1 Introdução	5
2.2 Terminologia Associada a Elevadores	5
2.3 Conforto Humano: Caso Geral	6
2.3.1 Variáveis Intrínsecas	7
2.3.2 Vibração como Variável Extrínseca	8
2.4 Conforto Humano: Viagem num Ascensor	17
2.4.1 Qualidade da Viagem	18
2.4.2 Melhoria da Qualidade da Viagem	19
2.5 Medição do Conforto em Ascensores	22
2.5.1 Características Fundamentais do Equipamento de Medição	22
2.5.2 Especificações para Efetuar as Medições	22
2.6 Instrumento para Medir o Conforto em Ascensores	23
3 Projeto e Métodos de Implementação	25
3.1 Introdução	25
3.2 Medição e Estimção de Variáveis Associadas ao Movimento	25
3.2.1 Posição	25
3.2.2 Acelerações e Vibrações	28
3.2.3 Velocidade e Navegação Inercial	29
3.3 Calibração de Sensor e Filtragem de Sinais	30
	xi

3.3.1	Calibração	31
3.3.2	Filtro de Kalman	32
3.4	Nível de Desempenho Ergonómico	35
3.5	Projeto do Sistema de Medição	36
3.5.1	Especificações Funcionais	36
3.5.2	Arquitetura Concetual	36
3.5.3	Especificações Tecnológicas	38
3.5.4	Especificações Operacionais	39
3.5.5	Hardware	40
3.6	Descrição Sumária do Elevador Protótipo	47
3.6.1	O Elevador Monta-Cargas	48
3.6.2	Especificações Funcionais, Tecnológicas e Operacionais	49
3.6.3	Arquitetura de Alto Nível	50
3.6.4	Arquitetura de Controlo	51
3.6.5	Lista de Pontos	52
3.7	Modelação do Comportamento do Variador de Velocidade do Elevador Modelo	52
3.8	Controlo do Elevador Modelo	55
3.8.1	Estrutura do Código	55
3.8.2	Perfis de Velocidade	56
4	Resultados Experimentais	63
4.1	Introdução	63
4.2	Resultados Obtidos para os Sensores	63
4.2.1	Acelerómetro	63
4.2.2	Barómetro	66
4.3	Resultados Obtidos pelo Sistema de Medição em Ascensores Públicos	66
4.3.1	Determinação de Aceleração, Velocidade e Posição	66
4.3.2	Inquérito Sobre o Desempenho Ergonómico	69
4.4	Teste dos Sensores no Modelo de Elevador	76
4.4.1	Sensor de Distância Ultrassónico	76
4.4.2	Sistema de Medição	78
4.5	Perfis Gerados	79
4.6	Discussão de Resultados	83
5	Conclusões e Trabalho Futuro	85
5.1	Conclusões	85
5.2	Trabalho futuro	85
	Bibliografia	87
A	Tarefas Utilizadas Quando a Posição Exata da Cabina Não é Conhecida	91
B	Tarefas Utilizadas Quando a Posição Exata da Cabina é Conhecida	95

C	Tarefas Adicionais	99
D	Código Arduino	101

Lista de Figuras

2.1	Fatores que contribuem para o desconforto [2]	7
2.2	Referenciais para avaliação de vibrações no corpo inteiro [4]	8
2.3	Modelo biomecânico teórico das frequências naturais do corpo humano [5] .	9
2.4	Sistema sujeito a uma força harmónica [3]	9
2.5	Representação gráfica da Equação (2.2) [3]	10
2.6	Ábaco com as curvas de ponderação em frequência principais [4]	12
2.7	Ábaco com as curvas de ponderação em frequência secundárias [4]	12
2.8	Curvas ideais (filtradas) de velocidade, aceleração e choque (derivada da aceleração) em função do tempo para uma viagem de ascensor entre dois patamares [11]	18
2.9	Curva-S para uma cabina de elevador [1]	20
2.10	Método mais comum de parar a cabina no patamar, adaptado [18]	21
2.11	Método de velocidade lenta, adaptado [18]	21
2.12	Posicionamento dos transdutores na cabina do ascensor [11]	23
2.13	EVA-625 [19]	23
3.1	Esquema de funcionamento do HC-SR04 , adaptado [20]	26
3.2	Sensor de distância ultrassónico HC-SR04	27
3.3	Placa GY-68 com sensor de pressão barométrico BMP180	28
3.4	Placa GY-521 com Unidade de Medição Inercial MPU6050	29
3.5	Consequência da integração do <i>erro de bias</i> , adaptado [27]	31
3.6	Tipos comuns de ruído	31
3.7	Arquitetura de alto nível do algoritmo do filtro de Kalman, adaptado [31] . .	33
3.8	Arquitetura de alto nível do SM	37
3.9	Arquitetura de baixo nível do SM	37
3.10	Sistema de menus utilizado no SM	39
3.11	Hardware do SM	40
3.12	Placa Arduino Mega	41
3.13	Placa de acrílico com sensores e chapas de aço	42
3.14	Cable doking station	42
3.15	Circuito de alimentação	43
3.16	Circuito de alimentação, esquema eletrónico	44
3.17	Interface SM-Homem	45
3.18	Circuito para navegação no menu	45

3.19	Esquema eletrónico para filtro PWM	46
3.20	Placa com três filtros passa baixo	47
3.21	Elevador modelo	47
3.22	Pormenores da caixa do elevador modelo	48
3.23	Pormenor das ampolas magnéticas no topo da cabina	48
3.24	Pormenores da casa das máquinas do elevador modelo	49
3.25	Consola HMI do variador de velocidade [34]	49
3.26	Arquitetura de alto nível do modelo de elevador	51
3.27	Arquitetura de controlo do elevador modelo	51
3.28	Variação da velocidade ao longo do tempo	53
3.29	Resposta do modelo ao degrau unitário	54
3.30	Resposta do modelo ao degrau	54
3.31	Resposta do modelo ao degrau	55
3.32	Representação gráfica das Tarefas 1, 2 e 3	56
3.33	Diagrama de blocos quando ativado o controlador PI	57
3.34	Perfil básico	57
3.35	Logograma para produzir o perfil básico	57
3.36	Perfil básico com velocidade de aproximação	58
3.37	Logograma para produzir o perfil básico com velocidade de aproximação	58
3.38	Perfil em rampa	58
3.39	Logograma para produzir o perfil em rampa	59
3.40	Perfil em rampa com velocidade de aproximação	59
3.41	Logograma para produzir o perfil em rampa com velocidade de aproximação	59
3.42	Logograma para produzir o perfil definido em função da posição	60
3.43	Anel de controlo para o controlo da velocidade em função da posição	60
3.44	Logograma para produzir o perfil com limitação de aceleração	61
3.45	Anel de controlo para o controlo da aceleração	61
4.1	Variação de <i>bias</i> devido à temperatura no eixo X	63
4.2	Variação de <i>bias</i> devido à temperatura no eixo Y	64
4.3	Variação de <i>bias</i> devido à temperatura no eixo Z	64
4.4	Resultado obtido no eixo Z após calibração	65
4.5	Integração dos resultados da Figura 4.4	65
4.6	Dupla integração dos resultados da Figura 4.4	65
4.7	Barómetro imóvel com as calibrações de fábrica	66
4.8	Trinta medições de aceleração utilizando acelerómetro	67
4.9	Trinta estimações de velocidade utilizando acelerómetro	67
4.10	Trinta estimações de posição utilizando acelerómetro	68
4.11	Trinta estimações de velocidade utilizando acelerómetro e barómetro	68
4.12	Trinta estimações de posição utilizando acelerómetro e barómetro	69
4.13	Perfil de aceleração do ascensor do edifício de civil na subida	70
4.14	Resultados da avaliação do conforto do ascensor do edifício de civil na subida	70
4.15	Perfil de aceleração do ascensor do edifício de civil na descida	71

4.16 Resultados da avaliação do conforto do ascensor do edifício de civil na descida	71
4.17 Perfil de aceleração do ascensor do edifício de matemática na subida	72
4.18 Resultados da avaliação do conforto do ascensor do edifício de matemática na subida	72
4.19 Perfil de aceleração do ascensor do edifício de matemática na descida	73
4.20 Resultados da avaliação do conforto do ascensor do edifício de matemática na descida	73
4.21 Perfil de aceleração do ascensor do edifício de eletrotécnica na subida	74
4.22 Resultados da avaliação do conforto do ascensor do edifício de eletrotécnica na subida	74
4.23 Perfil de aceleração do ascensor do edifício de eletrotécnica na descida	75
4.24 Resultados da avaliação do conforto do ascensor do edifício de eletrotécnica na descida	75
4.25 Pormenores relativos à instalação do sensor de ultrassons	76
4.26 Pormenor do sensor de ultrassons e placa de acrílico no contrapeso	77
4.27 Medição em centímetros do deslocamento da cabine em função do tempo em segundos	77
4.28 Instalação do SM na cabina do elevador modelo	78
4.29 Medição do deslocamento utilizando ultrassons e o SM	78
4.30 Medição da velocidade utilizando o variador de velocidade e o SM	79
4.31 Comparação entre a medição e receção da aceleração	79
4.32 Perfil 1, básico	80
4.33 Perfil 2, básico com velocidade de aproximação	80
4.34 Perfil 3, rampa	81
4.35 Perfil 4, rampa com velocidade de aproximação	81
4.36 Perfil 5, definido em função da posição	82
4.37 Perfil 6, com limitação de aceleração	82
4.38 Simulação de um salto na cabina utilizando o perfil 6	83

Lista de Tabelas

2.1	Variáveis intrínsecas relacionadas com o conforto [2]	6
2.2	Guia para a aplicação das curvas de ponderação em frequência principais e secundárias [4]	11
2.3	Parâmetros para as funções transferência das ponderações em frequência, tradução das tabelas A.1 e A.2 da norma ISO 2631-1 por Diogo Margarido [5] . .	13
2.4	Funções de transferência para um filtro digital de acordo com a norma ISO 2631-1 [4].	13
2.5	Escala de conforto presente na norma ISO 2631-1, anexo C [4]	17
2.6	Dinâmica da cabina de um ascensor para um edifício comercial [12]	19
2.7	Comparação da dinâmica da cabina de um ascensor entre edifícios residenciais e não residenciais [16]	19
2.8	Características dos transdutores segundo a ISO 18738 [11]	22
3.1	Diferença entre os modos de funcionamento do BMP180, retirado de [23] . . .	28
3.2	Resumo das especificações do MPU6050 [24]	29
3.3	NDE proposto	35
3.4	Lista de componentes	40
3.5	Especificações técnicas do Arduino MEGA [33]	41
3.6	Condições para identificação o patamar	51
3.7	Lista de pontos do TSX 3721	52
3.8	Parâmetros do modelo ARX(2,2,1) em função da tensão aplicada no variador de velocidade	54
4.1	Avaliação ergonómica dos seis perfis propostos	83

Lista de Siglas

ADC Conversor analógico-digital.

AmpOp Amplificador operacional.

AMZ600 módulo de entrada-saída analógica para o PLC.

ARX Modelo exógeno autoregressivo.

BMP180 Barômetro.

I²C Circuito Inter-integrado.

MEMS Sistema Micro-Eleto-Mecânico.

MPU6050 Acelerômetro.

NA Normalmente aberto.

NDE Nível de Desempenho Ergonômico.

NF Normalmente fechado.

PI Controlador proporcional e integral.

PLC Controlador lógico programável.

SM Sistema de Medição.

ST Texto estruturado.

Capítulo 1

Introdução

1.1 Contexto e Motivação

A espécie humana torna-se sedentária quando encontra satisfeitas as condições de sobrevivência e consequente propagação da espécie. De pequenas comunidades, avança ao longo dos séculos para as grandes cidades. A construção de habitações em altura tornou-se uma realidade e a solução para acolher cada vez mais habitantes. Se assim se resolvia um problema, criava-se outro: como levar os residentes para patamares elevados. A solução utilizada foram as escadas. Mas o Homem queria algo mais confortável, algo digno de um imperador. No século I a.C., em Roma, tal solução foi criada: o elevador. Com o passar dos séculos fazer o transporte vertical já não era suficiente. Era necessário que esse transporte fosse confortável. Este problema tem vindo a constituir-se como preocupação dos construtores de elevadores, tal como se constata nas palavras de Tou Wai Kei:

“The feeling of vibration and terrifying acceleration/deceleration greatly reduces the comfort of riding an elevator. In addition, those imperfections also give shocks to mechanical parts and shorten their life.” [1]

No seguimento desta ideia, pretende-se desenvolver um sistema de controlo que proporcione uma viagem ergonómica ao passageiro. O objetivo é que, desta forma, o passageiro não tenha perceção se a cabina se encontra a acelerar, desacelerar, em movimento ou em repouso, constituindo, por isso, uma mais-valia para o seu conforto.

No âmbito da tese de mestrado do aluno Nuno Ângelo foi construído um protótipo de um elevador monta-cargas. Este modelo encontra-se nas instalações da FCT-UNL, mais precisamente no laboratório 2.3 do Departamento de Engenharia Electrotécnica. Foi neste protótipo que foi desenvolvido e projetado um controlador que proporciona uma viagem ergonómica.

1.2 Objetivos

Para compreender os procedimentos necessários à construção de um controlador que assegure uma viagem ergonómica, foi necessário definir os seguintes objetivos:

- Identificar quais os valores de aceleração que causam conforto e desconforto nos ocupantes;
- Implementar um sistema de monitorização do comportamento do ascensor com interface para autómato programável (PLC);
- Projetar um sistema de controlo com otimização que garanta o conforto dos ocupantes nos movimentos de aceleração e de desaceleração de um ascensor.

Para dar resposta aos objetivos anteriores foram definidos os seguintes objetivos secundários:

- Criar o "Sistema de Medição" para medir as acelerações dos ascensores do quotidiano;
- Criar um sistema para medir e monitorizar a aceleração do protótipo de elevador;
- Definir um nível de conforto quantitativo para os passageiros.

O "Sistema de Medição" foi concebido para:

- Estimar o perfil de aceleração, velocidade e posição da cabina de um ascensor;
- Avaliar quantitativamente o desempenho ergonómico de um ascensor;
- Fornecer informação adicional ao controlador para melhorar o desempenho ergonómico da viagem.

1.3 Contribuições

Com esta dissertação fica disponível na Faculdade um dispositivo de fácil utilização e baixo custo que avalia o desempenho dos ascensores. Este dispositivo tem a designação de "Sistema de Medição", e os resultados são apresentados no decorrer da viagem, não sendo necessário futuro tratamento de resultados.

Este Sistema oferece a possibilidade de ser utilizado durante a manutenção ou a inspeção dos ascensores, indicando ao inspetor se existe a necessidade de reparar ou de inviabilizar a utilização do ascensor.

Para alcançar a viagem ergonómica no protótipo de elevador é proposto um controlador que apenas depende das acelerações sentidas na cabina e da distância ao patamar de destino.

É facultado ainda código para controlar o elevador protótipo. A sua utilização pode ter interesse para as cadeiras de Automação e Controlo. Pela forma como foi estruturado, os alunos podem, por exemplo, utilizar a parte do código que acerta a cabina num dado patamar sem ser necessário explicar-lhes quais as condições necessárias para atingir tal objetivo.

1.4 Estrutura da Dissertação

A presente dissertação encontra-se dividida em cinco capítulos. Seguidamente, e de modo esquemático, identificam-se os capítulos que constituem o presente trabalho e uma breve descrição dos conteúdos abordados:

Capítulo 1 - Introdução: considerações gerais a respeito do trabalho desenvolvido;

Capítulo 2 - Revisão do Estado da Arte: referências à literatura existente relacionada com o tema da dissertação;

Capítulo 3 - Projeto e Métodos de Implementação: apresentação dos objetivos propostos, projecto do Sistema de Medição e projeto de controlador;

Capítulo 4 - Resultados Experimentais: apresentação dos resultados obtidos;

Capítulo 5 - Conclusões: resumo do estudo apresentado, análise das vantagens e desvantagens e propostas para trabalhos futuros.

Capítulo 2

Revisão do Estado da Arte

2.1 Introdução

O estudo da resposta do corpo humano a vibrações é um tema pertinente, dado que as vibrações podem afetar a saúde humana. Estas podem causar alterações no sistema nervoso central, cardiovascular, respiratório, nos músculos ou mesmo na estrutura óssea. A vibração pode ainda causar desconforto, fadiga ou enjoo. No entanto, nem todo o tipo de vibrações é prejudicial ao Homem. Existem situações do quotidiano em que a questão das vibrações, mais ou menos intensas, exercem no indivíduo sensações de bem-estar, como a prática de desportos radicais (asa-delta, BTT ou mesmo Surf). Outras situações, como um simples aperto de mão, embalar um bebé ou mesmo uma massagem são disso também exemplo [2].

Neste capítulo veremos como as vibrações afetam o corpo Humano e a sua importância no projeto e utilização de ascensores.

2.2 Terminologia Associada a Elevadores

Para auxiliar o leitor na compreensão do uso de alguns termos mais técnicos associados a elevadores, entendeu-se necessário referenciá-los através da transcrição de partes do Decreto de Lei 513/70 de 30 de Outubro, secção Generalidade, sub-secção Definições, Artigo 3.º:

2. *Ascensor*. — Elevador destinado ao transporte de pessoas ou de pessoas e carga e cuja cabina tem dimensões e constituição que permitem o acesso de pessoas.

5. *Cabina*. — Órgão do elevador onde são transportadas as pessoas ou a carga.

6. *Caixa*. — Local onde se desloca a cabina ou a cabina e o contrapeso.

8. *Casa das máquinas*. — Local destinado à máquina de tracção da cabina e aos aparelhos de comando.

9. *Contrapeso*. — Órgão destinado a equilibrar o peso da cabina e de parte da sua carga.

12. *Elevador*. — Instalação destinada ao transporte de pessoas ou carga entre níveis definidos de serviço numa cabina que se desloca ao longo de guias verticais ou ligeiramente inclinadas sobre a vertical.

13. *Guias*. — Órgãos destinados a guiar o movimento da cabina e do contrapeso.

15. *Máquina de tracção*. — Máquina que movimenta a cabina.

19. *Patamar*. — Pavimento ou plataforma onde a cabina estaciona para entrada e saída de pessoas ou carga.

23. *Roda de aderência*. — Roda que por atrito movimenta os cabos de suspensão.

2.3 Conforto Humano: Caso Geral

No século XIV Bernardino Ramazzini (médico italiano) desenvolveu um estudo sobre lesões associadas ao trabalho. O interesse sobre esta temática estendeu-se ao período da Revolução Industrial. Todavia, só após a Segunda Grande Guerra surgiram estudos mais aprofundados, como os de K.F.H. Murrell, fundador da Ergonomic Research Society, cujo objetivo era estudar a adaptação do trabalho ao Homem. Outros estudos e iniciativas têm contribuído para o desenvolvimento e entendimento do impacto das vibrações no organismo humano, destacando-se Griffin com *Handbook of Human Vibration* [2]. Nesta publicação, Griffin esquematizou algumas das causas mais comuns de desconforto em veículos, como se observa na Figura 2.1.

Na Figura 2.1 são identificados, pelo autor, dois grupos de variáveis: intrínsecas (ao centro) e extrínsecas (do lado esquerdo). À direita são identificados alguns dos efeitos causados por estas variáveis. As primeiras relacionam-se com o meio no qual o indivíduo se insere, enquanto as segundas estão relacionadas com o próprio indivíduo. As variáveis intrínsecas (Tabela 2.1) remetem para a ideia que o conforto varia ao longo do dia e da vida de um indivíduo.

Intra-subjetiva	Inter-subjetiva
Postura	Altura e massa
Posição corporal	Género e idade
Orientação corporal	Condição física
	Resposta dinâmica corporal

Tabela 2.1: Variáveis intrínsecas relacionadas com o conforto [2]

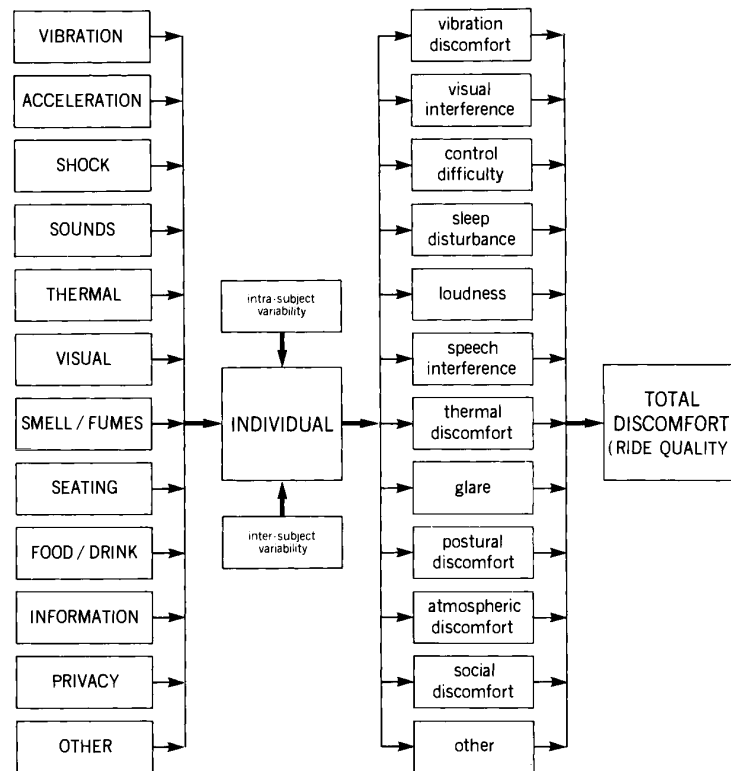


Figura 2.1: Fatores que contribuem para o desconforto [2]

2.3.1 Variáveis Intrínsecas

As variáveis intrínsecas dividem-se em intra-subjetivas e inter-subjetivas. As variáveis intra-subjetivas remetem para a ideia que a forma como o indivíduo se expõe às vibrações tem impacto na sua sensação de desconforto. Nas variáveis inter-subjetivas, a sensação de conforto varia consoante a massa da cabeça, do tronco, das ancas e das pernas, aliada ao carácter elástico-amortecedor do pescoço, coluna, abdómen e pernas [3]. Vejamos em seguida as variáveis intra-subjetivas apresentadas na Tabela 2.1.

No caso específico de uma pessoa em pé, se esta mudar de uma postura de pernas esticadas para uma postura de joelhos dobrados consegue, diminuir drasticamente o desconforto gerado pelas vibrações. Esta mudança de postura permite ao organismo absorver parte da vibração. A posição corporal refere-se ao modo como a pessoa se encontra: sentada, em pé ou deitada. Consoante a posição, a superfície corporal exposta às vibrações é diferente, com consequências na percepção que o indivíduo tem de conforto.

Griffin [2] faz referência aos estudos de Parsons sobre a orientação corporal, apresentando a seguinte situação: quando uma pessoa está sentada e sujeita a vibrações, a sensação de desconforto causada pelo encosto da cabeça varia se a vibração for aplicada lateralmente ou de trás para a frente¹.

A orientação corporal necessita de um referencial que depende da posição corporal. Esta dependência explica-se pela necessidade da origem do referencial ter que coincidir com

¹ Os estudos de Parsons mostram que é menos desconfortável se a vibração for aplicada lateralmente.

a superfície pela qual as vibrações são transmitidas ao corpo humano. A Organização Internacional para Padronização concebeu a norma ISO 2631-1 na qual propõe os referenciais representados na Figura 2.2 em função da posição corporal.

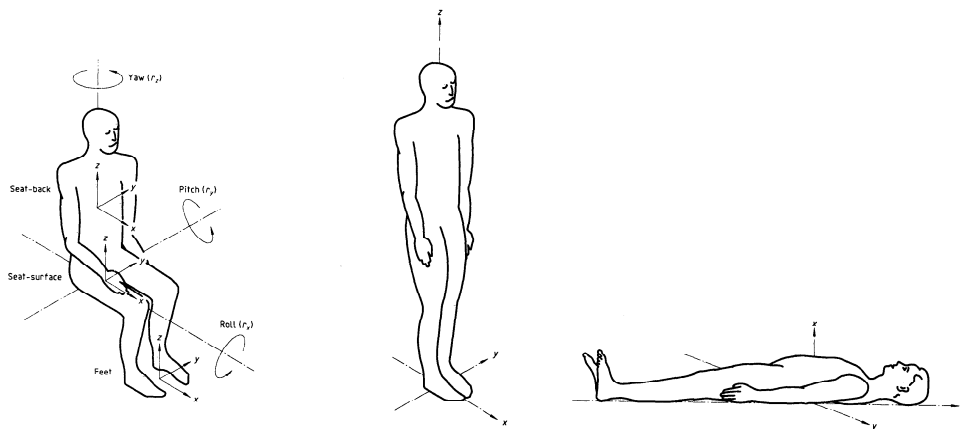


Figura 2.2: Referenciais para avaliação de vibrações no corpo inteiro [4]

2.3.2 Vibração como Variável Extrínseca

No estudo da vibração como variável extrínseca existem vários parâmetros a considerar, dos quais se destacam: frequência, duração e intensidade.

Frequência da Vibração

O corpo humano possui vários órgãos e ossos, cada um com a sua frequência natural de oscilação. Nesse sentido, foram elaborados vários estudos biomecânicos por forma a perceber quais as frequências naturais do corpo humano. A Figura 2.3 explicita que existe uma vasta gama de frequências naturais ao longo do corpo humano. Por norma, estudam-se frequências entre 0.1 e 100 Hz.

Quando a frequência da vibração de um sistema coincide com a sua frequência natural, este começa a oscilar com grandes amplitudes. Este fenómeno designa-se por ressonância. A ressonância é atenuada caso o sistema tenha amortecedores. Para explicar melhor este conceito, tomemos como o exemplo o sistema esquematizado na Figura 2.4. Este sistema é constituído por uma massa (m), uma mola (k) e um amortecedor (c). O sistema desloca-se segundo o eixo x devido a uma força harmónica sinusoidal $F(t)$ [3].

A Equação (2.1) representa o equilíbrio dinâmico do sistema. Partindo desta equação é possível chegar à Equação (2.2). A última representa a amplitude a que o sistema estaria sujeito, se a força F_0 agisse estaticamente. A Figura 2.5 é a representação gráfica da Equação (2.2).

$$m\ddot{x} + c\dot{x} + kx = F(t) = F_0 \cos(\omega t) \quad (2.1)$$

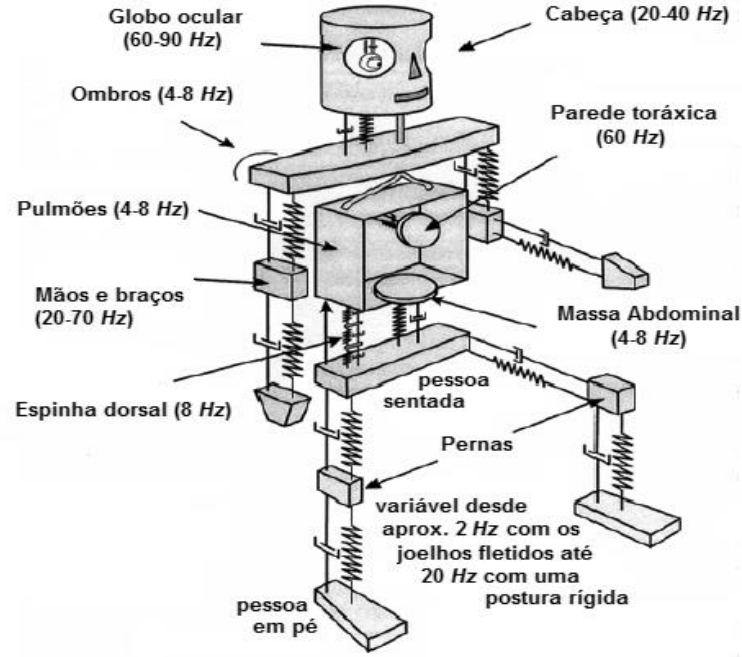


Figura 2.3: Modelo biomecânico teórico das frequências naturais do corpo humano [5]

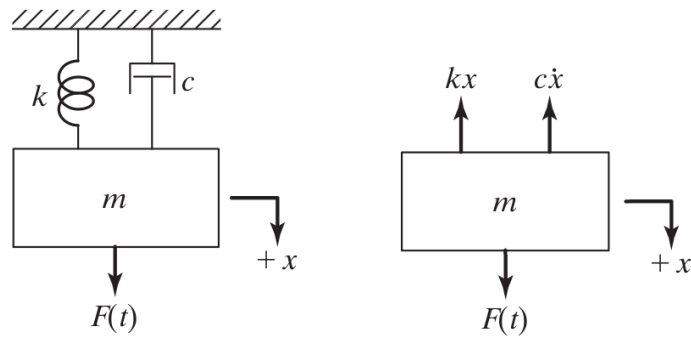


Figura 2.4: Sistema sujeito a uma força harmônica [3]

$$M = \frac{X}{\frac{F_0}{k}} = \frac{X}{\delta_{est}^*} = \sqrt{(1 - r^2)^2 + (2\zeta r)^2} \quad (2.2)$$

onde:

- M factor de amplificação;
- X amplitude máxima;
- F_0 força estática aplicada;
- k constante de rigidez da mola;
- δ_{est}^* deformação estática;
- r relação entre a frequência de oscilação (ω) e a frequência natural (ω_n);
- ζ fator de amortecimento.

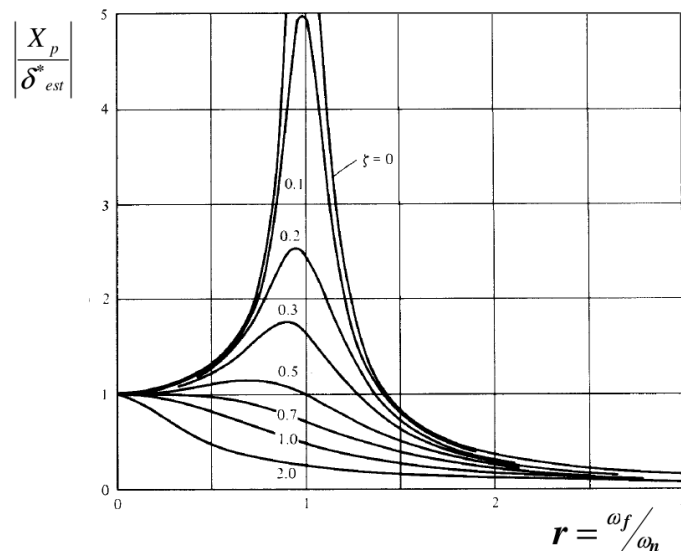


Figura 2.5: Representação gráfica da Equação (2.2) [3]

Observando a Figura 2.5, verifica-se que a amplitude do movimento (eixo vertical) atinge o máximo quando ocorre o fenômeno de ressonância ($r = 1$). Observa-se também que, devido ao fator de amortecimento do sistema, a amplitude máxima na ressonância diminui. Consoante a frequência de oscilação, assim existirá um aumento ou decréscimo das forças de extensão/compressão aplicadas à estrutura óssea e aos órgãos internos. Se um sistema permanecer em ressonância durante um certo período de tempo, existe a possibilidade de colapsar. Na sequência do exposto resulta uma sensação de desconforto e, em última análise, problemas de saúde associados à frequência de vibração.

Bruno Sergio [6] de acordo com o pensamento de Griffin [2] identifica os riscos causados pela frequência da vibração na saúde do Homem, os quais se destacam:

- a) “Pessoas expostas à vibração de corpo inteiro na faixa de 1 a 30 Hz podem ter influência em seu sistema nervoso e muscular, o qual pode apresentar a diminuição dos reflexos e dificuldades posturais”;
- b) “Pessoas expostas à vibração de corpo inteiro em frequências inferiores a 20 Hz podem ter influência em seu sistema cardiovascular, o qual pode apresentar um aumento da frequência cardíaca”;
- c) “Pessoas expostas à vibração de corpo inteiro na faixa de 1 a 10 Hz, com intensidades próximas a 4.9 m/s^2 , podem ter influência em seu sistema respiratório, o qual pode alterar o controle da respiração e ventilação pulmonar”;
- d) “Pessoas expostas à vibração de corpo inteiro na faixa de 0.1 a 0.7 Hz podem apresentar influências em diversos sistemas, os quais podem causar enjoos, náuseas, perda de peso, redução da acuidade visual, insónia, desordens no intestino e cólicas no cólon”.

Uma das possíveis conclusões a retirar do exposto é a seguinte: a frequência afeta consideravelmente o conforto e a saúde humana. Deste modo, para determinar se uma frequência é mais ou menos prejudicial ao corpo humano são utilizados fatores ponderativos. Existem duas formas de os aplicar: ou recorrendo a ábacos, ou utilizando filtros digitais. Começemos pela utilização dos ábacos.

Para utilizar os ábacos (Figuras 2.6 e 2.7) é necessário escolher uma curva de ponderação em frequência. A escolha é feita consoante a informação disponibilizada na Tabela 2.2.

Frequency weighting	Health	Comfort	Perception	Motion sickness
W_k	z-axis, seat surface	z-axis, seat surface z-axis, standing vertical recumbent (except head) x-, y-, z-axes, feet (sitting)	z-axis, seat surface z-axis, standing vertical recumbent	—
W_d	x-, y-axis, seat surface	x-, y-axis, seat surface x-, y-axes, standing horizontal recumbent y-, z-axes, seat-back	x-, y-axis, seat surface x-, y-axes, standing horizontal recumbent	—
W_f	—	—	—	vertical
W_c	x-axis, seat-back	x-axis, seat-back	x-axis, seat-back	—
W_e	—	r_x -, r_y -, r_z -axes, seat	r_x -, r_y -, r_z -axes, seat	—
W_j	—	vertical recumbent (head)	vertical recumbent (head)	—

Tabela 2.2: Guia para a aplicação das curvas de ponderação em frequência principais e secundárias [4]

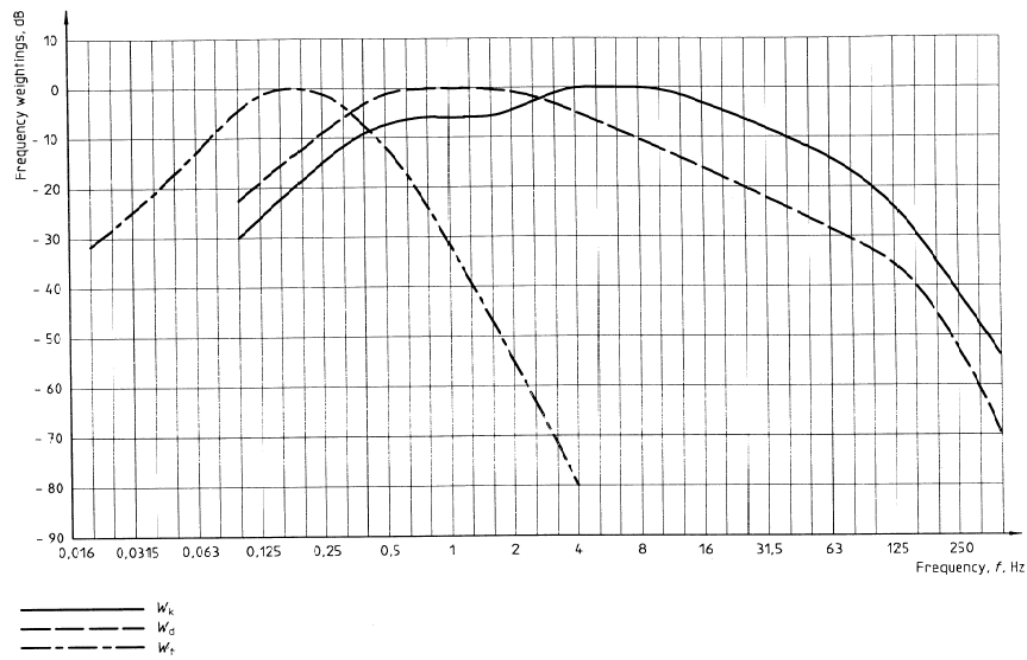


Figura 2.6: Ábaco com as curvas de ponderação em frequência principais [4]

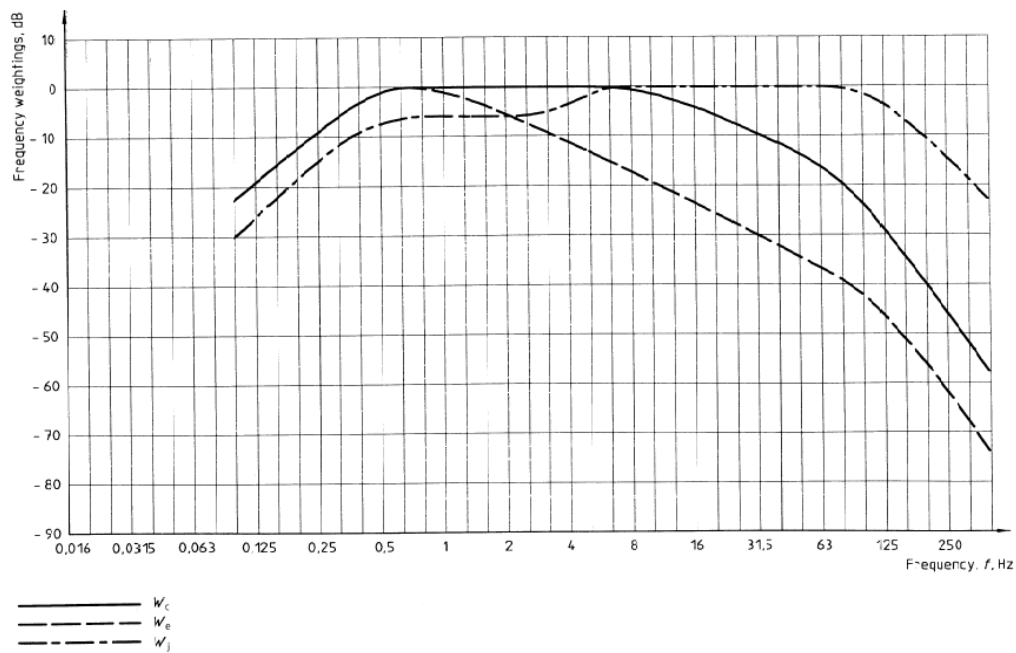


Figura 2.7: Ábaco com as curvas de ponderação em frequência secundárias [4]

Para projetar os filtros digitais é necessário recorrer à Tabela 2.3, a qual indica os parâmetros a introduzir na funções de transferência do filtro digital (Equação (2.3)).

O filtro digital é expresso pela Equação (2.3).

$$H(z) = H_h(z) \cdot H_l(z) \cdot H_t(z) \cdot H_s(z) \quad (2.3)$$

onde:

- $H_h(z) \cdot H_l(z)$ representa os limites das bandas passantes;
 $H_t(z) \cdot H_s(z)$ representa uma curva de ponderação em frequência;
 $H_t(z) = 1$ para W_j ;
 $H_s(z) = 1$ para W_c, W_d e W_e .

Ponderação	Limites de Banda		Transição aceleração- velocidade			Passo ascendente			
	f_1 (Hz)	f_2 (Hz)	f_3 (Hz)	f_4 (Hz)	Q_4	f_5 (Hz)	Q_5	f_6 (Hz)	Q_6
W_k	0,4	100	12,5	12,5	0,63	2,37	0,91	3,35	0,91
W_d	0,4	100	2,0	2,0	0,63	∞	--	∞	--
W_f	0,08	0,63	∞	0,25	0,86	0,0625	0,8	0,1	0,8
W_c	0,4	100	8,0	8,0	0,63	∞	--	∞	--
W_e	0,4	100	1,0	1,0	0,63	∞	--	∞	--
W_j	0,4	100	∞	∞	--	3,75	0,91	5,32	0,91

Tabela 2.3: Parâmetros para as funções transferência das ponderações em frequência, tradução das tabelas A.1 e A.2 da norma ISO 2631-1 por Diogo Margarido [5]

De acordo com a norma ISO 2631-1 é necessário recorrer a quatro funções de transferência para definir o filtro. As quatro funções de transferência, já simplificadas, estão representadas na Tabela 2.4.

Limite de banda passa alto	$ H_h(z) = \sqrt{\frac{f^4}{f^4 + f_1^4}}$
Limite de banda passa baixo	$ H_l(z) = \sqrt{\frac{f_2^4}{f^4 + f_2^4}}$
Transição da aceleração-velocidade	$ H_t(z) = \sqrt{\frac{f^2 + f_3^2}{f_3^2}} \cdot \sqrt{\frac{f_4^4 \cdot Q_4^2}{f_4^4 \cdot Q_4^2 + f^2 \cdot f_4^2 (1 - 2Q_4^2) + f_4^2 \cdot Q_4^2}}$
Passo ascendente	$ H_s(z) = \frac{Q_6}{Q_5} \cdot \sqrt{\frac{f^4 \cdot Q_5^2 + f^2 \cdot f_5^2 (1 - 2Q_5^2) + f_5^4 \cdot Q_5^2}{f^4 \cdot Q_5^2 + f^2 \cdot f_5^2 (1 - 2Q_5^2) + f_6^4 \cdot Q_6^2}}$

Tabela 2.4: Funções de transferência para um filtro digital de acordo com a norma ISO 2631-1 [4].

Duração da Vibração

A duração da vibração e o número de horas de vibração acumulada têm impacto na saúde humana. Refira-se a título de exemplo, o síndrome das vibrações mão-braço, que resulta do uso prolongado de ferramentas vibratórias. Esta doença depende não só da duração das vibrações, mas também, de acordo com o Decreto-Lei nº 46/2006, do “*nível de vibrações produzidas pela máquina ou ferramenta, (...), da temperatura no espaço do posto de trabalho, do método de trabalho e da ergonomia das tarefas profissionais*” [7]. Existe uma probabilidade de 10% de um indivíduo contrair este síndrome ao ser exposto a uma vibração diária de 2.5 m/s^2 durante doze ou mais anos [7].

Para evitar impactos severos na saúde, o Decreto-Lei nº 46/2006 impõe que a Equação (2.4) seja cumprida. Esta equação define o limite de exposição diária de vibração para o corpo inteiro² de uma pessoa durante oito horas de trabalho, e indica que devem ser tomadas medidas preventivas caso o valor ultrapasse 0.5 m/s^2 [7].

$$A(8) = k a_w \sqrt{\frac{T}{t_0}} < 1.15 \text{ m/s}^2 \quad (2.4)$$

onde:

- a_w aceleração eficaz ponderada [m/s^2];
- T duração diária total da exposição às vibrações [s];
- t_0 duração de referência de oito horas (28 800 segundos);
- k fator multiplicativo.

Miwa em 1968 chegou à conclusão que o desconforto aumenta até um valor limite e que após esse valor não é possível ficar “mais desconfortável” [2]. Os resultados de Miwa foram confirmados por um estudo levado a cabo por uma equipa de investigação da NASA em 1978 [8]. Neste estudo é relatado que após passar o limite do desconforto, este foi diminuindo com o tempo. Uma das hipóteses apresentada pela equipa da NASA é que conforme o tempo aumenta, o corpo humano começa a adaptar-se à vibração. Contudo Ykä Marjanen (2010) afirma que “*It is logical to assume some form of time dependency where discomfort will increase as a function of time.*” [9]. No período que medeia as investigações de Miwa e Ykä, Griffin e Whitham propuseram o método designado por *Root-Mean-Qua* representado pela Equação 2.5, e que associa o conforto à duração, intensidade e frequência. No entanto o resultado da equação não tem significado físico, porque vem expresso em $\text{m/s}^{1.75}$.

$$r.m.q = \left(\int_0^T a_w(t)^4 dt \right)^{0.25} \quad (2.5)$$

² Para o caso do sistema mão-braço o limite é 5 m/s^2 e as medidas preventivas devem ser inicializadas caso exceda os 2.5 m/s^2 .

Intensidade

A norma ISO 2631-1:1997 *Mechanical Vibration and Shock – Evaluation of Human Exposure to Whole Body Vibration – Part 1: General Requirements* pretende associar a intensidade da vibração à saúde, ao conforto e ao enjoo.

Com o objetivo de avaliar a severidade da vibração, esta norma apresenta o Método Básico para efetuar a avaliação. O método consiste na utilização da aceleração r.m.s. (*Root-Mean-Square*) ponderada. O cálculo do valor da aceleração r.m.s ponderado deve ser elaborado de acordo com a Equação (2.6).

$$a_{w_{ef}} = \left(\frac{1}{T} \int_0^T a_w(t)^2 dt \right)^{0.5} \quad (2.6)$$

onde:

$a_{w_{ef}}$ aceleração r.m.s. ponderada [m/s^2];
 $a_w(t)$ aceleração ponderada instantânea [m/s^2];
 T duração da medição [s].

Uma vez que o Método Básico pode não ser apropriado para descrever os efeitos da vibração no corpo humano a norma define um Fator de Crista (Equação (2.7)). Se o Fator de Crista for menor ou igual a nove, então a utilização da aceleração r.m.s. ponderada é suficiente para descrever os efeitos da vibração no corpo humano. Caso contrário torna-se necessário utilizar métodos adicionais. Os métodos adicionais como, por exemplo, r.m.s. em Contínuo ou a Dose de Vibração de Quarta Potência (VDV) [4].

$$\text{Fator de Crista} = \left| \frac{\text{aceleração de pico máxima}}{\text{aceleração r.m.s. ponderada}} \right| = 9 \quad (2.7)$$

O Método r.m.s. em contínuo tem em consideração os choques ocasionais e o seu cálculo é feito de acordo com a Equação (2.8a). A Equação (2.8a) pode ser, de acordo com a norma ISO 8041, aproximada à Equação (2.8b). A diferença de resultados entre estas duas equações é pequena para choques de curta duração [10]. Com este método é possível definir o *Valor Máximo da Vibração Transiente* (MTVV) (Equação (2.9)) e condições da sua utilização serão mencionada mais adiante.

$$a_{w_{ef}(t_0)} = \left(\frac{1}{\tau} \int_{t_0-\tau}^{t_0} a_w(t)^2 dt \right)^{0.5} \quad (2.8a)$$

$$a_{w_{ef}(t_0)} = \left(\frac{1}{\tau} \int_{-\infty}^{t_0} a_w(t)^2 \exp \left[\frac{t-t_0}{\tau} \right] dt \right)^{0.5} \quad (2.8b)$$

onde:

$a_{w_{ef}}(t_0)$	aceleração r.m.s. ponderada em contínuo [m/s^2];
$a_w(t)$	aceleração r.m.s. ponderada instantânea [m/s^2];
τ	tempo de integração para a média em contínuo [s];
t	tempo (variável de integração) [s];
t_0	tempo de observação (tempo instantâneo) [s].

$$MTVV = \max[a_{w_{ef}}(t_0)] \quad (2.9)$$

A Dose de Vibração de Quarta Potência é o método sugerido por Griffin e Whitham: *Método Root-Mean-Qua*. Este método, apesar de ter merecido algumas críticas pela comunidade científica, pelo facto de não ter significado físico, foi adotado para avaliar os efeitos da vibração [2]. O seu cálculo é feito pela Equação (2.5).

Em termos práticos quando se procede à avaliação da influência das vibrações no corpo humano, o Método Básico tem sempre de ser utilizado e os seus valores apresentados num relatório. Contudo, podendo não ser suficiente, deve ser utilizado um método adicional e os resultados deste devem constar do relatório de avaliação [4]. O método adicional é escolhido conforme os rácios nas Equações (2.10) ou (2.11) sejam excedidos.

$$\frac{MTVV}{a_w} = 1.5 \quad (2.10)$$

$$\frac{VDV}{a_w T^{0.25}} = 1.75 \quad (2.11)$$

Para avaliar o impacto da vibração, devem ser combinadas as vibrações segundo os três eixos de um referencial ortogonal. A Equação (2.12) alude a esta combinação e o seu resultado representa o “valor de vibração total” aplicada no corpo.

$$a_w = \sqrt{(k_x^2 \times a_{wx}^2 + k_y^2 \times a_{wy}^2 + k_z^2 \times a_{wz}^2)} \quad (2.12)$$

Na Equação (2.12) os parâmetros a_{wx} , a_{wy} e a_{wz} representam as acelerações segundo x, y e z com as respetivas ponderações aplicadas (Tabela 2.2). k_x , k_y e k_z são fatores ponderativos para cada uma das direções que, para o caso de uma pessoa em pé, têm o valor 1 [4].

No Anexo C da norma ISO 2631-1 é fornecida informação que associa a intensidade da vibração ao conforto de um passageiro em transportes públicos, a qual foi transposta para a Tabela 2.5.

Valor de Vibração Total (m/s^2)	Nível de Conforto
< 0.315	Não é desconfortável
0.315 – 0.63	Um pouco desconfortável
0.5 – 1	Razoavelmente desconfortável
0.8 – 1.6	Desconfortável
1.25 – 2.5	Muito desconfortável
> 2	Extremamente desconfortável

Tabela 2.5: Escala de conforto presente na norma ISO 2631-1, anexo C [4]

2.4 Conforto Humano: Viagem num Ascensor

Como apresentado na secção 2.3, existem diversos fatores que afetam o conforto humano, sendo a vibração o fator que ganha maior preponderância. No caso dos ascensores estuda-se a velocidade, as vibrações, a aceleração e o nível de pressão sonora.

Começemos pela velocidade. A velocidade da cabina relaciona-se com o conforto da pessoa apenas pelo facto da viagem demorar mais ou menos tempo. O corpo humano, caso esteja privado da visão, apenas tem perceção que se está a deslocar caso exista uma variação de velocidade.

Quanto aos órgãos sensoriais, o ouvido é o que mais importância tem no estudo do conforto humano numa viagem num ascensor. Ou seja, é o único órgão sensorial que recebe estímulos relacionados com o movimento.

O ouvido humano tem a capacidade de detetar pequenas variações de pressão. Consoante a diferença entre a pressão dentro e fora do ouvido, os tímpanos são forçados a vibrar com frequências diferentes. Conforme a vibração do tímpano, o cérebro pode interpretar como som, ruído ou variação de altura. Sendo que a maior parte dos ascensores efectua pequenos percursos, a variação de pressão que importa considerar é aquela que está relacionada com o ruído. Se a cabina tiver um mau isolamento sonoro, e o seu deslocamento produzir ruído, o passageiro ir-se-à sentir incomodado.

O ouvido humano consegue detetar, ainda, acelerações angulares (detetada nos canais semicirculares) e acelerações lineares (detetada pelos órgãos otolíticos). Quer os canais, quer os órgãos localizam-se no ouvido interno e fazem parte do aparelho vestibular. Existem três canais semicirculares e dois órgãos otólitos (utrículo e sáculo) em cada ouvido. Relativamente aos órgãos otólitos, o primeiro deteta a aceleração no plano horizontal, enquanto o segundo deteta as acelerações no plano vertical. Uma pessoa tem perceção

de acelerações lineares a partir de $(0.0015 \pm 0.0005) \text{ m/s}^2$ [4].

Mas a aceleração não é apenas sentida no ouvido. Consoante a sua intensidade e frequência pode ser sentida noutras partes do corpo, ou mesmo no corpo inteiro, como foi referido anteriormente.

2.4.1 Qualidade da Viagem

A norma ISO 18738 apresenta a Figura 2.8 como o percurso ideal de ascensor. Observa-se que existe um período inicial em que a velocidade aumenta até atingir um valor máximo, o qual é mantido durante a maior parte do percurso. Quando a cabina se aproxima do patamar de chegada, a sua velocidade diminui até cessar o movimento da mesma.

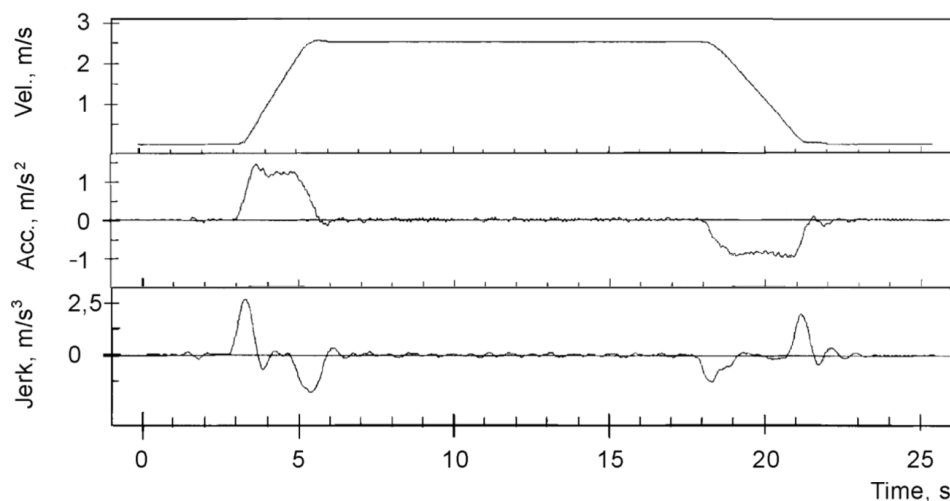


Figura 2.8: Curvas ideais (filtradas) de velocidade, aceleração e choque (derivada da aceleração) em função do tempo para uma viagem de ascensor entre dois patamares [11]

A norma ISO 18738 não apresenta valores limites de referência para nenhum destes parâmetros. Barney, em *Elevator Traffic Handbook*, por exemplo, apresenta valores do limiar de conforto para a aceleração e choque de 1.5 m/s^2 e 2 m/s^3 [12] assim como Howkins [13]. Outros autores apresentam valores máximos de 1.2 m/s^2 e 4.5 m/s^3 [14] ou ainda 1 m/s^2 e 4 m/s^3 [15]. Retomando Howkins, um valor de choque superior a 6 m/s^3 é inaceitável para o público geral e um valor superior a 0.7 m/s^3 em elevadores hospitalares também é inaceitável uma vez que pode gerar problemas aos doentes que tenham sido sujeitos a intervenções cirúrgicas. Note-se que na Tabela 2.5 estes valores são classificados como pertencentes ao nível “Desconfortável”.

As Tabelas 2.6 e 2.7 apresentam valores de referência para a velocidade da cabina, indicando que quanto maior for o percurso, maior será a velocidade da cabina.

Distância percorrida [m]	Velocidade máxima [m/s]	Aceleração máxima [m/s ²]
< 20	< 1.00	0.4
20	1.00	0.4 - 0.7
32	1.60	0.7 - 0.8
50	2.50	0.8 - 0.9
63	3.15	1.0
100	5.00	1.2 - 1.5
120	6.00	1.5
> 120	> 6.00	1.5

Tabela 2.6: Dinâmica da cabina de um ascensor para um edifício comercial [12]

Percurso [m]	Velocidade máxima [m/s]	
	Edifícios não residenciais	Edifícios residenciais
< 17	0.50 - 1.00	0.75 - 1.00
18 - 29	1.00 - 1.75	
30 - 44	1.75 - 2.50	1.00 - 1.50
45 - 59	2.50 - 3.50	1.25 - 2.00
60 - 74	3.50 - 4.00	1.75 - 2.50
75 - 89	4.00 - 5.00	2.50 - 5.50
90 - 150	5.00 - 6.00	-
> 150	6.00 - 8.00	

Tabela 2.7: Comparação da dinâmica da cabina de um ascensor entre edifícios residenciais e não residenciais [16]

2.4.2 Melhoria da Qualidade da Viagem

Os ascensores foram sujeitos a evoluções tecnológicas ao longo dos anos. Atualmente, os construtores de ascensores debatem-se com problemas de paragem precisa da cabina no patamar, e de redução da intensidade das acelerações e vibrações. A forma mais eficaz de resolver este problema reside na utilização de controladores. Com o auxílio destes, é possível assegurar uma viagem rápida (tempo de deslocação entre patamares), precisa (desfasamento menor que 3 mm), confortável para o utilizador e um aumento do tempo

de vida dos constituintes do elevador (choques mais reduzidos) [1]. Um dos requisitos fundamentais dos ascensores topo de gama é limitar as vibrações da cabina, recorrendo à monitorização e controlo da velocidade ao longo da viagem [17].

Se o controlador for um PLC ((*Programmable Logic Controller*)), então as suas funções principais são [18]:

- Gerir as chamadas do elevador ao patamar;
- Identificar e guardar os patamares por onde a cabina passe;
- Controlar a velocidade da cabina;
- Calcular a posição da cabina;
- Controlar o travão da máquina de tração;
- Garantir as funções de segurança (ex. não ultrapassar excesso de carga e garantir possibilidade de paragem de emergência);

Controlo de Velocidade e Chegada ao Patamar

O perfil de velocidade mais eficaz para a cabina é uma curva-S (Figura 2.9). Este otimiza o conforto do passageiro, uma vez que o arranque e a paragem são bastante suaves.

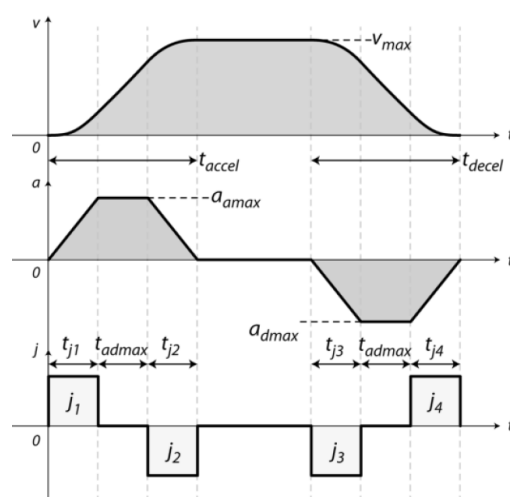


Figura 2.9: Curva-S para uma cabina de elevador [1]

Os ascensores que existem no quotidiano têm duas formas distintas de acertar a cabina ao patamar. A primeira forma, mais económica e utilizada na maioria dos elevadores, é recorrer a sensores que indicam quando se inicia a redução de velocidade (Sensor 1, Figura 2.10) e quando cessa o movimento (Sensor 2, Figura 2.10).

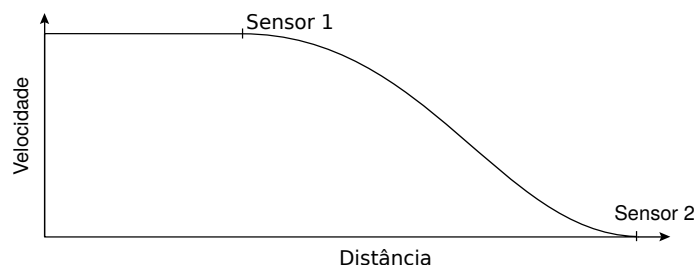


Figura 2.10: Método mais comum de parar a cabina no patamar, adaptado [18]

Muitas das empresas de elevadores optam por introduzir um sensor adicional (Figura 2.11) para garantir uma paragem mais precisa [18]. Quando o Sensor 1 é acionado, a cabina é forçada a deslocar-se com uma velocidade de aproximação. A velocidade mantém-se constante até chegar ao Sensor Adicional, a partir desse ponto inicia-se o processo de paragem.

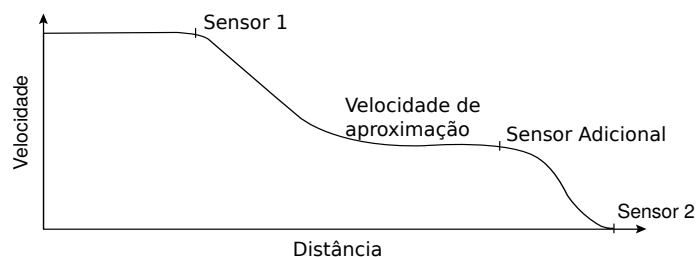


Figura 2.11: Método de velocidade lenta, adaptado [18]

A segunda forma, mais dispendiosa, e por norma encontrada em elevadores que servem um elevado número de patamares, consiste em registar a posição atual da cabina e a posição de destino. Fazendo a diferença entre as duas posições, o controlador pode reduzir a velocidade da cabina de modo a obter uma desaceleração suave. Idealmente, a velocidade e aceleração tendem para zero à medida em que a cabina se aproxima do patamar, conseguindo uma paragem suave e impercetível para o passageiro [1].

Esta técnica permite menores tempos de viagem relativamente aos métodos anteriores. Para a utilização deste método é necessário conhecer exatamente onde se encontra a cabina. A técnica mais utilizada para determinar a posição da cabina é o acoplamento de um *encoder* óptico ao veio da máquina de tração³. Esta técnica de posicionamento da cabina não está isenta de erros, uma vez que o cabo que transmite a força da máquina de tração para a cabina pode escorregar, originando um desfasamento de alguns milímetros entre a posição real da cabina e a posição medida pelo *encoder*. [1]

³Existem ascensores que utilizam sensores de distância ultrassónicos em vez do *encoder*.

2.5 Medição do Conforto em Ascensores

2.5.1 Características Fundamentais do Equipamento de Medição

De acordo com a norma ISO 18738: *Lifts (elevators) — Measurement of lift ride quality*, são utilizados transdutores capazes de interpretar sinais representativos da vibração (segundo os três eixos de um referencial ortogonal) e do nível de pressão sonora. A Tabela 2.8 resume as principais características dos transdutores utilizados.

	Vibração	Aceleração	Pressão sonora
Alcance mínimo	$-1.5 - 1.5 \text{ m/s}^2$	$7 - 13 \text{ m/s}^2$	30 – 90 dB
Resolução	0.005 m/s^2	0.01 m/s^2	1 dB
Band Limit	0.5 – 100 Hz	Filtro passa baixo de 10 Hz (Butterworth com 2 pólos)	N/A

Tabela 2.8: Características dos transdutores segundo a ISO 18738 [11]

2.5.2 Especificações para Efetuar as Medições

O processamento dos dados deve começar pelo menos meio segundo antes da porta começar a fechar (no patamar de partida) e deve acabar pelo menos meio segundo depois da porta começar a abrir (no patamar de chegada). A recolha dos dados deve começar antes e depois deste intervalo. A norma ISO 18738 refere que deve ser feita, pelo menos, uma medição de subida e de descida entre o primeiro e o último patamares do ascensor.

Os transdutores utilizados para medir as vibrações devem ser colocados no chão da cabina a uma distância máxima de cem milímetros do seu centro. No caso do transdutor destinado a medir o nível de pressão sonora, este deve ser colocado a (1.5 ± 0.1) metros sobre a região circular anterior e alinhado com o sentido positivo do eixo x.

Os eixos dos transdutores têm que estar alinhados com os do referencial (Figura 2.12).

Para a medição de vibrações, o transdutor tem que ser fixado ao chão da cabina. A norma sugere que a sua fixação seja feita por uma pressão de pelo menos sessenta kilopascal, uma vez que esta é a pressão, aproximadamente, exercida por um pé humano [11].

Na tentativa de não adulterar o resultado das medições, não podem frequentar o ascensor mais do que duas pessoas, mantendo-se em silêncio, quietas e colocadas de modo a não desequilibrar a cabina. De igual modo os pés têm de estar afastados pelo menos cento e cinquenta milímetros do transdutor destinado a medir vibrações e a trezentos milímetros do equipamento destinado a medir a pressão sonora.

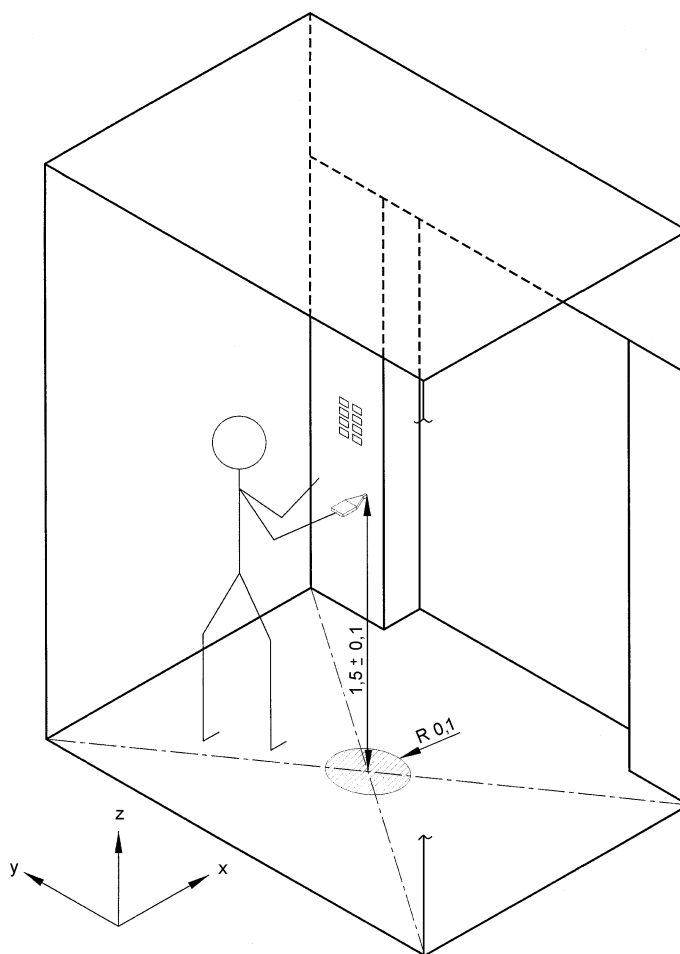


Figura 2.12: Posicionamento dos transdutores na cabina do ascensor [11]

2.6 Instrumento para Medir o Conforto em Ascensores

O *Elevator Vibration Analysis* ou EVA-625 (Figura 2.13) é o sistema mais utilizado para avaliar a qualidade da viagem em ascensores, escadas e esteiras rolantes. Este aparelho é considerado o *standard* industrial para proceder à avaliação da qualidade da viagem num elevador e mede: velocidade, acelerações, vibrações, choques máximos e nível de pressão sonora. O software que acompanha o dispositivo tem ferramentas de diagnóstico como análise em frequência (FFT) que permitem identificar se os carris estão desalinhados e desenha as curvas relativas ao movimento [19].



Figura 2.13: EVA-625 [19]

O EVA-625 tem muitos aspetos positivos, mas tem um preço elevado e não avalia quantitativamente o conforto nas viagens de ascensores.

Capítulo 3

Projeto e Métodos de Implementação

3.1 Introdução

Neste capítulo são apresentadas as metodologias utilizadas para responder aos objetivos definidos no capítulo 1.

O presente capítulo está dividido em três partes. A primeira diz respeito às metodologias aplicadas para medir e estimar a aceleração, a velocidade e a posição de uma cabina de elevador. É referido, ainda, o procedimento necessário para efetuar a calibração de um acelerómetro, técnicas de filtragem de sinais e é apresentada uma proposta de avaliação ergonómica.

A segunda parte inicia-se com o projeto do Sistema de Medição, no qual são aplicados os conceitos apresentados no capítulo 2 e na primeira parte do capítulo 3. Por fim é apresentado o modelo de elevador.

3.2 Medição e Estimação de Variáveis Associadas ao Movimento

Na presente dissertação pretende-se conhecer o deslocamento da cabina ao longo do percurso. Com esta informação é possível exercer um melhor controlo da velocidade, que se reflete na melhoria do conforto para o passageiro. Para determinar a posição da cabina serão estudados três métodos distintos, cada qual utilizando um tipo de sensor. Os sensores são: sensor de distância ultrassónico, barómetro e acelerómetro.

3.2.1 Posição

Sensor de Distância Ultrassónico

Este tipo de sensor envia e recebe sinais ultrassónicos. Dependendo do tempo entre a emissão e a receção do ultrassom é possível determinar distâncias (Equação (3.1)). A

Figura 3.1 esquematiza o funcionamento.

$$d = \frac{v}{2} \cdot \Delta t \quad (3.1)$$

onde:

d	distância do objeto ao sensor ultrassônico	$[m];$
Δt	intervalo de tempo entre a emissão e recepção do som	$[s].$
v	velocidade do som no ar	$340 \text{ m/s};$

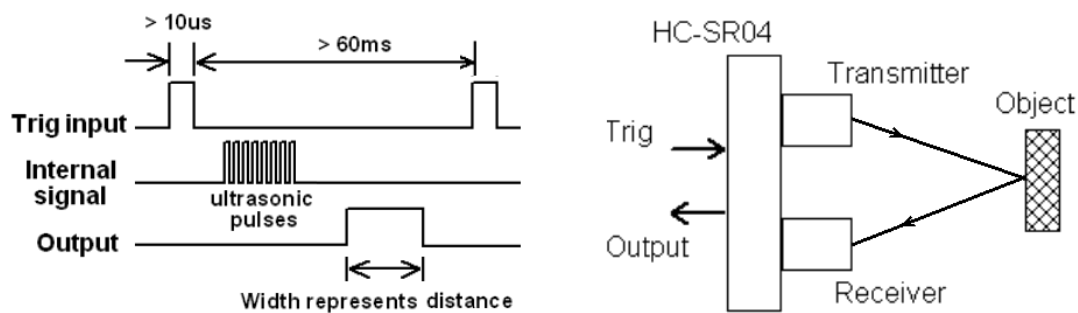


Figura 3.1: Esquema de funcionamento do HC-SR04 , adaptado [20]

- No sensor o canal *Trig* recebe um valor HIGH durante pelo menos $10 \mu s$;
- São emitidos 8 pulsos a 40 kHz contra o objeto em estudo;
- O Output (ou Echo) passa de LOW para HIGH;
- O som é refletido pelo objeto e volta ao sensor;
- O Output passa de HIGH para LOW;
- Consoante o tempo em que o Output esteve em HIGH é possível determinar a distância a que o objeto se encontra do sensor [21].

O sensor utilizado é o HC-SR04 (Figura 3.2). Este sensor consegue medir distâncias entre 2 cm e 4 m com uma resolução de 0.3 cm. Os ultrassons produzidos pelo sensor têm uma frequência de 40 kHz e são projetados segundo um ângulo de 30°. Para o seu correto funcionamento é necessário uma tensão de 5 Volt, uma corrente de 15 mA e a área mínima para a reflexão do som deve ser 0.5 m^2 [21].

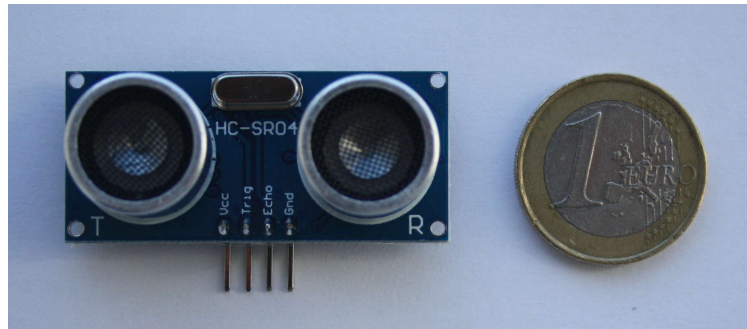


Figura 3.2: Sensor de distância ultrassónico HC-SR04

Barómetro

O barómetro mede a pressão que a coluna de ar exerce no sensor e recorrendo, à Equação 3.2, consegue determinar a altura a que o sensor se encontra [22].

$$z = \frac{T_0}{\alpha} \left(1 - \left(\frac{p}{p_0} \right)^{\frac{R\alpha}{gM}} \right) \quad (3.2)$$

onde:

z	altitude	$[m]$
p	pressão ambiente à altitude z	$[hPa]$
T_0	temperatura ao nível médio do mar	$288.15 K$
α	gradiente térmico	$0.0065 K/m$;
p_0	pressão ao nível médio do mar	$1013.25 hPa$;
R	constante dos gases ideais	$8.314472 N \cdot m/mol \cdot K$;
g	constante de gravidade terrestre	$9.80665 m/s^2$;
M	massa molar do ar	$0.0289644 kg/mol$.

Note-se que os parâmetros p_0 e T_0 variam de dia para dia e ao longo do mesmo. Contudo, para o tempo de uma viagem de elevador, estes são constantes.

O barómetro utilizado é o BMP180, que utiliza tecnologia MEMS (*Sistema Micro-Electro-Mecânico*) para efetuar as medições. O sensor consegue detetar pressões no intervalo de 300 a 1100 hPa. A tensão de alimentação deve situar-se no intervalo compreendido entre 1.62 e 3.6 Volt. No que toca à corrente consumida, esta varia entre 3 a 32 μA , podendo atingir picos de 650 μA quando procede a conversões. A variação anterior está relacionada com a resolução do BMP180. Quanto maior for a resolução, mais corrente é necessária consumir. Não é apenas o consumo de corrente que está relacionado com a resolução, mas também o ruído e a frequência de aquisição de dados, como é elucidado na Tabela 3.1. O BMP180 tem instalado um ADC que permite codificar o sinal analógico¹ em 16 bits, ou caso se utilize o modo de maior resolução: 19 bits. A comunicação com

¹ Associado à medição da pressão

outros dispositivos é feita por I²C. Durante a comunicação, o valor LOW pode variar entre 0 e 0.324 Volt e o HIGH entre 1.296 e 1.62 Volt [23]. Este protocolo está implementado na biblioteca de Korneliusz Jarzebski: *BMP085.h*.

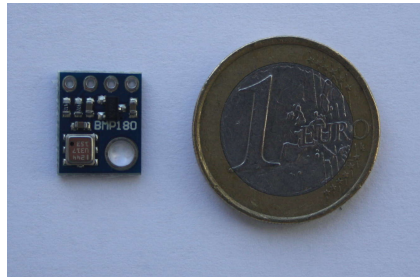


Figura 3.3: Placa GY-68 com sensor de pressão barométrico BMP180

MODE	Internal samples	Conversion time [ms]	Current [μ A]	RMS noise [hPa]	RMS noise [m]
Ultra low power	1	4.5	3	0.06	0.5
Standard	2	7.5	5	0.05	0.4
High resolution	4	13.5	7	0.04	0.3
Ultra high resolution	8	25.5	12	0.03	0.25
Advanced resolution	-	76.5	32	0.02	0.17

Tabela 3.1: Diferença entre os modos de funcionamento do BMP180, retirado de [23]

3.2.2 Acelerações e Vibrações

Acelerómetro

As acelerações e as vibrações são medidas através de acelerómetros. Os acelerómetros utilizam os princípios das Leis de Newton para determinar acelerações. Recorrem, por isso, à Equação (3.3) para determinar acelerações ou vibrações.

$$\vec{F} = m \cdot \vec{a} \quad (3.3)$$

onde:

- \vec{F} força aplicada;
- m massa do corpo;
- \vec{a} aceleração sofrida.

O acelerómetro utilizado é o MEMS MPU6050 instalado na placa GY-521 (Figura 3.4). O MPU6050 é constituído por três acelerómetros e três giroscópios, alinhados segundo os eixos desenhados na placa(Figura 3.4), razão para ser considerado uma Unidade de Medição Inercial. O MPU6050 possui ainda um ADC de 16 bits e um sensor de temperatura.

Algumas das especificações são apresentadas na Tabela 3.2. O MPU6050 funciona com 2.375 a 3.46 Volt e a corrente necessária, com todas as funcionalidades ativas, é 3.9 mA. No caso de se ativar apenas o acelerómetro, a corrente necessária é de 500 μ A. A comunicação com outros dispositivos é feita por I²C. Durante a comunicação o valor LOW pode variar entre -0.5 e 1.038 Volt e o HIGH entre 1.663 e 3.96 Volt [24]. Este protocolo está implementado na biblioteca de Jeff Rowberg: *MPU6050.h*.

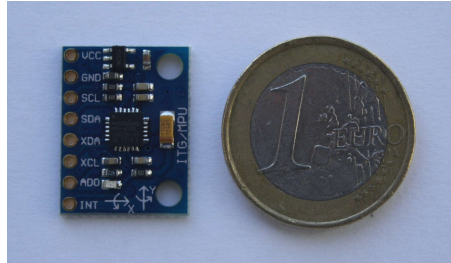


Figura 3.4: Placa GY-521 com Unidade de Medição Inercial MPU6050

PARAMETER	TYP	Units
ACCELEROMETER		
Full-Scale Range	± 4	g
Output data rate	1000	Hz
Zero-G Level Change vs. Temperature (axies)	± 35 (X, Y) ± 60 (Z)	mg
Noise Spectral Density	400	$\mu g / \sqrt{Hz}$
GYROSCOPE		
Full-Scale Range	± 250	$^{\circ}/s$
Output data rate	8000	Hz
Zero-rate output Variation Over Temperature	± 20	$^{\circ}/s$
Noise Spectral Density	0.005	$^{\circ}/s / \sqrt{Hz}$
TEMPERATURE SENSOR		
Range	-40 to +85	$^{\circ}C$
Sensitivity	340	LSB/ $^{\circ}C$

Tabela 3.2: Resumo das especificações do MPU6050 [24]

3.2.3 Velocidade e Navegação Inercial

Recorrendo aos sensores apresentados, existem duas hipóteses de estimar a velocidade: derivar a posição ou integrar a aceleração. Pelo facto das medições fornecidas pelos sensores terem ruído associado, dificulta a obtenção do correto valor de velocidade. Ao proceder à derivação de ruído, o ruído é intensificado e a informação relativa à velocidade perde-se. Na integração de ruído, a velocidade aumenta/diminui com a passagem do tempo.

Sendo possível utilizar qualquer um dos métodos, torna-se necessário filtrar o sinal do sensor. Os filtros utilizados para obter bons resultados, a partir da derivação, têm uma complexidade espacial e temporal muito superior aos filtros utilizados para integração. Assim, a velocidade será estimada com base na aceleração (Equação (3.4)).

$$a(t) = \frac{dv}{dt} = \frac{d^2p}{dt^2} \Rightarrow \begin{cases} v(t) = v_0 + a(t)\Delta t \\ p(t) = p_0 + v_0\Delta t + \frac{a(t)}{2}\Delta t^2 \end{cases} \quad (3.4)$$

onde:

- a aceleração;
- v velocidade;
- v_0 velocidade inicial;
- p posição;
- p_0 posição inicial;
- Δt intervalo de tempo.

A técnica para estimar a velocidade de um veículo (ou mesmo a sua posição e/ou orientação) recorre apenas a forças inerciais e é conhecida por Navegação Inercial, a qual requer uma Unidade de Medição Inercial para ser utilizada (por exemplo o MPU6050).

A grande vantagem da Navegação Inercial consiste na possibilidade de navegar em territórios desconhecidos, não necessitando de referenciais externos nem de comunicar com outros sistemas. A Unidade de Medição Inercial utiliza os acelerómetros para determinar a intensidade das forças aplicadas e o princípio da conservação do momento angular para determinar a orientação do corpo (recorrendo aos giroscópios), estimado-se assim a orientação, a posição e a velocidade do corpo [25] [26].

3.3 Calibração de Sensor e Filtragem de Sinais

Os resultados provenientes de sensores vêm sempre com ruído associado, sendo este mais ou menos intenso consoante a qualidade do sensor. Na tentativa de obter um melhor sinal são efetuadas compensações e aplicados filtros. Tomemos o exemplo apresentado na Figura 3.5. Nesta são apresentadas as leituras efetuadas por um acelerómetro. O sensor deveria ler aceleração nula, mas por não ter sido feito o zero, o sensor origina leituras incorretas. Este tipo de erro é conhecido como *erro de bias*. Como se pode observar na figura, após a primeira integração, o sensor estaria a aumentar a sua velocidade a um ritmo constante. Atente-se ao gráfico mais à direita no qual a aceleração é integrada duas vezes com o intuito de estimar a posição. O acelerómetro encontra-se parado, mas observando apenas o gráfico, este deslocou-se alguns metros ou até quilómetros. O *erro de bias* pode também variar consoante a temperatura.

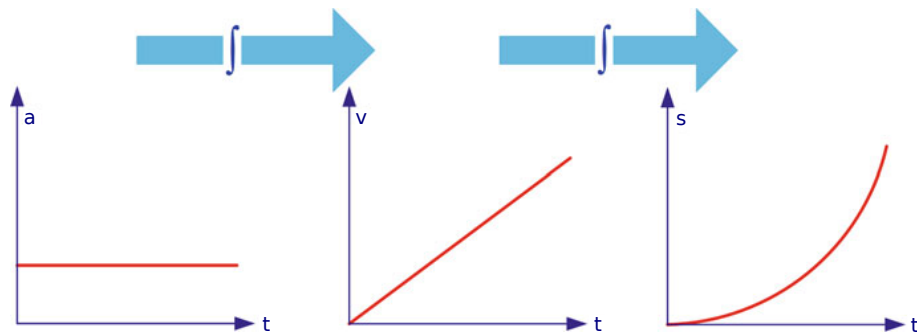
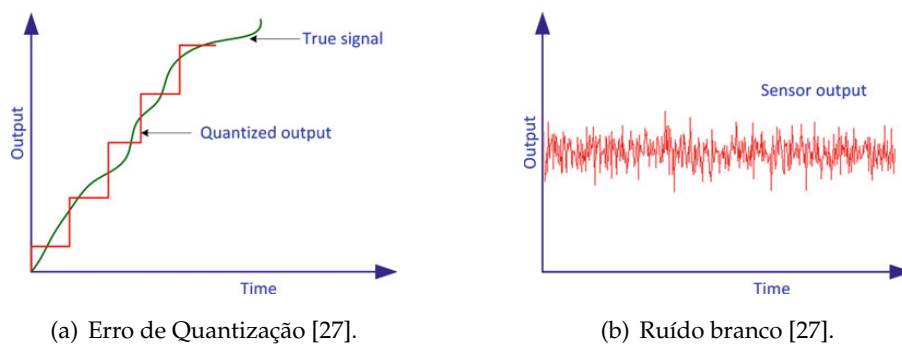


Figura 3.5: Consequência da integração do erro de *bias*, adaptado [27]

Existem outros tipos de erros associados a sensores, como por exemplo o causado pela conversão de sinal analógico para digital (Figura 3.6(a)), ou mesmo ainda os causados pela fonte de alimentação e pelas características intrínsecas dos semicondutores do sensor (Figura 3.6(b)) [26].



(a) Erro de Quantização [27].

(b) Ruído branco [27].

Figura 3.6: Tipos comuns de ruído

3.3.1 Calibração

O processo de calibração consiste em associar a um valor de tensão uma grandeza física. Os sensores utilizados vêm calibrados de fábrica, contudo será efetuada a calibração do acelerómetro para obter melhores resultados. Para a calibração do acelerómetro é utilizado o *Six-Position Static Test* (Teste Estático de Seis Posições), que fornece bons resultados e é extremamente fácil de executar. O teste consiste em montar o acelerómetro numa superfície nivelada e durante um período de dez a quinze minutos são recolhidos dados. Em seguida, o sensor é virado ao contrário de modo a que o eixo aponte no sentido oposto, recolhendo dados durante o mesmo período de tempo. O código utilizado para fazer a calibração encontra-se no Apêndice D.

É necessário realizar o procedimento anterior para cada um dos três eixos. Aos resultados recolhidos aplicam-se as Equações (3.5a) e (3.5b) para determinar o *bias* e o fator de escala para cada eixo. Após calcular estes dois valores utiliza-se a Equação (3.5c) para

converter o valor medido pelo acelerómetro em m/s^2 [28].

$$b_a = \frac{f_{cima} + f_{baixo}}{2} \quad (3.5a)$$

$$S_a = \frac{f_{cima} - f_{baixo}}{R \cdot g} \quad (3.5b)$$

$$a_{rel} = \frac{valor_{med} - b_a}{S_a} \quad (3.5c)$$

onde:

$valor_{med}$	valor medido pelo acelerómetro ;
a_{rel}	aceleração relativa;
b_a	<i>bias</i> associado ao acelerómetro;
S_a	fator de escala;
f_{cima}	média dos dados quando o eixo aponta para cima ;
f_{baixo}	média dos dados quando o eixo aponta para baixo;
R	Alcance máximo da medição (nesta dissertação assume o valor 4);
g	aceleração da gravidade;

3.3.2 Filtro de Kalman

Utiliza-se, frequentemente, em variadas aplicações o filtro de Kalman. Sendo uma poderosa ferramenta capaz de estimar o estado de um sistema dinâmico estocástico a partir de medições com ruído, a sua utilização é adequada a sensores. Este filtro permite resultados muito próximos da realidade, mesmo que as medições dos sensores fiquem temporariamente indisponíveis [29] [30]. Como foram utilizados sensores de baixo custo foi utilizado o Filtro de Kalman nesta dissertação.

Para a utilização do filtro de Kalman é necessário começar por escrever (no espaço de estados) as equações que representam o sistema em estudo (Equação (3.6)) e as medições efetuadas (Equação (3.7)).

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \quad (3.6)$$

$$z_k = Hx_k + v_k \quad (3.7)$$

onde:

$x \in \mathbb{R}^{n \times 1}$	variáveis de estado;
$A \in \mathbb{R}^{n \times n}$	estado do processo;
$B \in \mathbb{R}^{n \times l}$	relaciona o valor de controlo com o estado;
$u \in \mathbb{R}^{l \times n}$	variáveis de controlo;
$w \in \mathbb{R}^{n \times n}$	ruído associado ao processo;
$z \in \mathbb{R}^{m \times 1}$	variáveis medidas;
$H \in \mathbb{R}^{m \times n}$	relaciona o estado com a medição;
$v \in \mathbb{R}^{m \times 1}$	ruído associado à medição.

Este filtro permite lidar com ruídos: um associado ao processo e outro à medição do sensor, w e v , respectivamente. O filtro assume que estes ruídos são ruído branco, têm uma distribuição de probabilidades normal² e são independentes um do outro.

$$p(w) \sim N(0, Q)$$

$$p(v) \sim N(0, R)$$

O filtro de Kalman consiste num ciclo com duas etapas. São elas: Atualização Temporal e a Atualização pela Medição.

Na primeira é feita uma previsão temporal das estimativas do vetor de estado e da matriz de covariância com base na informação do processo no instante anterior ($k-1$). Enquanto que na segunda é efetuada uma correção das estimativas com base na leitura mais recente (k). Em termos práticos, o algoritmo a implementar é apresentado na Figura 3.7.

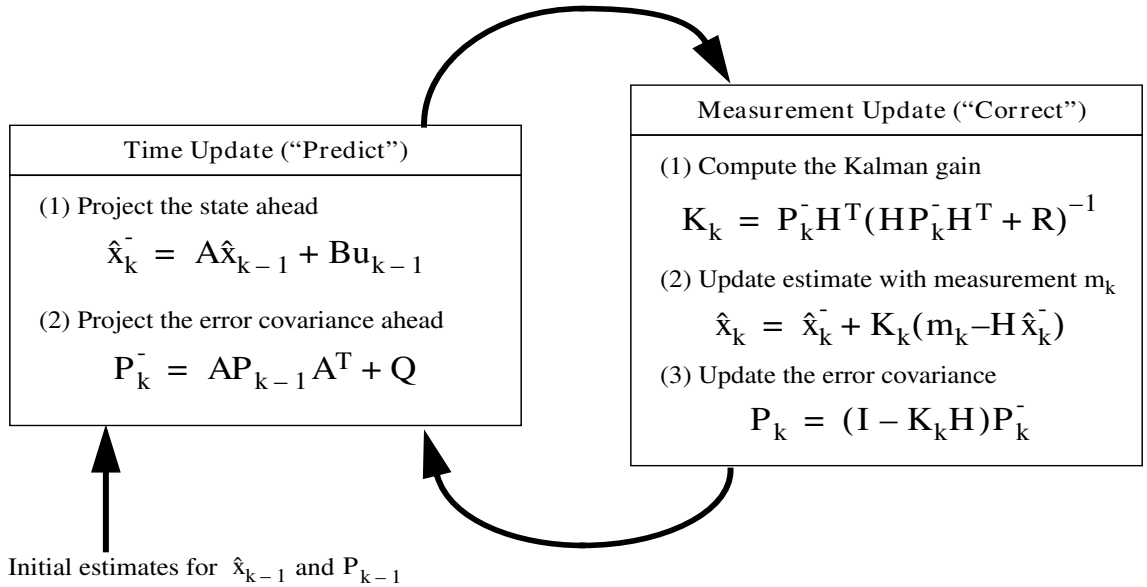


Figura 3.7: Arquitetura de alto nível do algoritmo do filtro de Kalman, adaptado [31]

Filtro de Kalman Orientado para Navegação Inercial

Vejamos agora como é que o filtro de Kalman foi utilizado no trabalho recorrendo apenas a um acelerómetro e à técnica de navegação inercial. Sendo que o estudo será realizado num ascensor, foram tomadas as seguintes considerações:

- O chão da cabina é perpendicular à direção do movimento;
- O acelerómetro é colocado no chão da cabina a uma distância máxima de 100 mm do seu centro;
- O acelerómetro permanece paralelo ao chão da cabina durante a viagem;

² As covariâncias podem variar ao longo do tempo.

A matriz A e o vetor \hat{x} são definidos com base nas equações do movimento (Equação (3.4)):

$$A = \begin{bmatrix} 1 & \Delta T & \Delta T^2/2 \\ 0 & 1 & \Delta T \\ 0 & 0 & 1 \end{bmatrix} \quad \hat{x} = \begin{bmatrix} p \\ v \\ a \end{bmatrix}$$

A matriz de estado do processo (A) e a de covariância do ruído do processo (Q) estão relacionadas. O estado do processo consiste em sucessivas integrações de aceleração deste modo, o ruído do processo ocorre devido às integrações. As equações do movimento apresentadas partem do pressuposto que a aceleração é constante. A matriz Q é calculada da seguinte forma [32], [28]:

$$Q = \int_0^{\Delta T} e^{D\tau} Q_t e^{D^T\tau} d\tau \quad (3.8)$$

Onde:

$$e^{D\tau} = I + D\tau + \frac{D^2\tau^2}{2!} + \frac{D^3\tau^3}{3!} + \dots \quad (3.9)$$

D representa a matriz de transição que é definida como:

$$\dot{X} = DX \quad \Leftrightarrow \quad \begin{bmatrix} \dot{x} \\ \dot{v} \\ 0 \end{bmatrix} = D \begin{bmatrix} x \\ v \\ \ddot{x} \end{bmatrix} \quad \Rightarrow \quad D = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

A matriz Q_t representa a covariância do ruído do processo em tempo contínuo:

$$Q_t = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & q_c \end{bmatrix}$$

Fazendo as respectivas substituições na Equação 3.8 obtém-se:

$$Q = \int_0^{\Delta T} \begin{bmatrix} 1 & \tau & \frac{1}{2}\tau^2 \\ 0 & 1 & \tau \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & q_c \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ \tau & 1 & 0 \\ \frac{1}{2}\tau^2 & \tau & 1 \end{bmatrix} d\tau = q_c \begin{bmatrix} \frac{1}{20}\Delta T^5 & \frac{1}{8}\Delta T^4 & \frac{1}{6}\Delta T^3 \\ \frac{1}{8}\Delta T^4 & \frac{1}{3}\Delta T^3 & \frac{1}{2}\Delta T^2 \\ \frac{1}{6}\Delta T^3 & \frac{1}{2}\Delta T^2 & \Delta T \end{bmatrix}$$

O vetor m e as matrizes H e R são definidos de acordo com o sensor utilizado, que para o caso é o acelerómetro. A matriz H indica a grandeza que se pretende medir; no vetor m é colocado o valor medido pelo sensor e na matriz R é representado o ruído associado ao sensor:

$$H = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \quad m = \begin{bmatrix} \text{aceleração medida} \end{bmatrix} \quad R = \begin{bmatrix} \sigma_{\text{acelerómetro}}^2 \end{bmatrix}$$

Existe a necessidade de inicializar a matriz de covariância dos erros (P_{k-1}) e o vetor das variáveis de estado (\hat{x}_{k-1}) para o filtro funcionar corretamente. Os valores da matriz P_{k-1} devem ser elevados, enquanto que os valores de \hat{x}_{k-1} devem corresponder à realidade. Estas imposições garantem que a solução final convirjam para o resultado real. Saliente-se que os elementos destas matrizes variam a cada iteração e devem ser reinicializados quando se pretende efetuar novas medições.

$$P_{k-1} = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix} \quad \hat{x}_{k-1} = \begin{bmatrix} p_0 \\ 0 \\ 0 \end{bmatrix}$$

A implementação do filtro encontra-se no Apêndice D.

3.4 Nível de Desempenho Ergonómico

Tendo por base o anexo C da norma ISO 18738 é proposta uma relação entre intensidade de aceleração e um Nível de Desempenho Ergonómico (NDE). Foram atribuídos incrementos de 0.3 m/s^2 entre níveis. O nível intermédio é definido para o intervalo 0.6 a 1 m/s^2 porque neste intervalo a norma associa os termos “Um pouco desconfortável” e “Razoavelmente desconfortável”. A partir de 1.25 m/s^2 a norma avalia como “Muito desconfortável”.

O NDE será atribuído consoante o resultado da aceleração r.m.s. (Equação 3.10a) e também é aplicável à aceleração r.m.q. (Equação 3.10b). A razão para utilizar r.m.s. resulta da obrigatoriedade expressa na norma. O r.m.q é utilizado por ter sido desenvolvido por Griffin e Whitham e ter sido aceite como o melhor método para avaliar vibrações.

Consoante o intervalo a que pertença o valor da aceleração, independentemente do método, assim será atribuído um NDE (Tabela 3.3). Pretende-se que desta forma seja possível atribuir uma nota quantitativa do conforto humano em ascensores.

$a_{r.m.*} [m/s^2]$	NDE
< 0.3	5
$0.3 - 0.6$	4
$0.6 - 1.0$	3
$1.0 - 1.3$	2
> 1.3	1

$$a_{r.m.s.} = \left(\frac{\sum_{i=1}^N a_{x_i}^2 + a_{y_i}^2 + a_{z_i}^2}{N} \right)^{0.5} \quad (3.10a)$$

$$a_{r.m.q.} = \left(\frac{\sum_{i=1}^N a_{x_i}^4 + a_{y_i}^4 + a_{z_i}^4}{N} \right)^{0.25} \quad (3.10b)$$

Tabela 3.3: NDE proposto

Para estudar a aplicabilidade deste NDE foi efetuado um inquérito a trinta estudantes da FCT-UNL. Neste inquérito foi pedido que andassem em três ascensores distintos e os avaliassem numa escala de 1 a 5, tendo apenas em conta vibrações e acelerações. Os resultados deste inquérito encontram-se no capítulo 4.

3.5 Projeto do Sistema de Medição

Com base nas metodologias apresentadas até ao momento, pode-se iniciar a apresentação do Sistema de Medição (SM), sendo que este tem como objetivos:

- estimar perfis da cabina de um ascensor;
- avaliar quantitativamente o desempenho ergonómico de um ascensor;
- auxiliar no controlo de um ascensor;

3.5.1 Especificações Funcionais

As especificações funcionais do SM são as seguintes:

- Aquisição de dados para desenhar os perfis de aceleração, velocidade e posição;
- Avaliar quantitativamente numa escala de 1 a 5 o desempenho ergonómico de um ascensor;
- Enviar os dados recolhidos para um autómato programável para ser possível controlar o conforto na cabina do ascensor.

3.5.2 Arquitetura Concetual

A Figura 3.8 representa a arquitetura de alto nível do SM. Esta figura tem um código de cores associado, o qual é também utilizado na arquitetura de baixo nível (Figura 3.9). A saber:

- Cinzento* — Encapsulamento do SM;
- Verde escuro* — Troca de informação entre o SM e o utilizador;
- Branco* — Unidade de processamento de dados;
- Verde claro (C)* — Sistema capaz de receber/interpretar sinais analógicos;
- Vermelho* — Barómetro;
- Vermelho claro (A)* — Pressão atmosférica e temperatura;
- Azul escuro* — Acelerómetro e giroscópio;
- Azul claro (B)* — Forças inerciais e temperatura;
- Amarelo torrado* — Componentes internos do SM (Figura 3.9 apenas);
- Setas de maior espessura* — Comunicação por I²C (Figura 3.9 apenas).

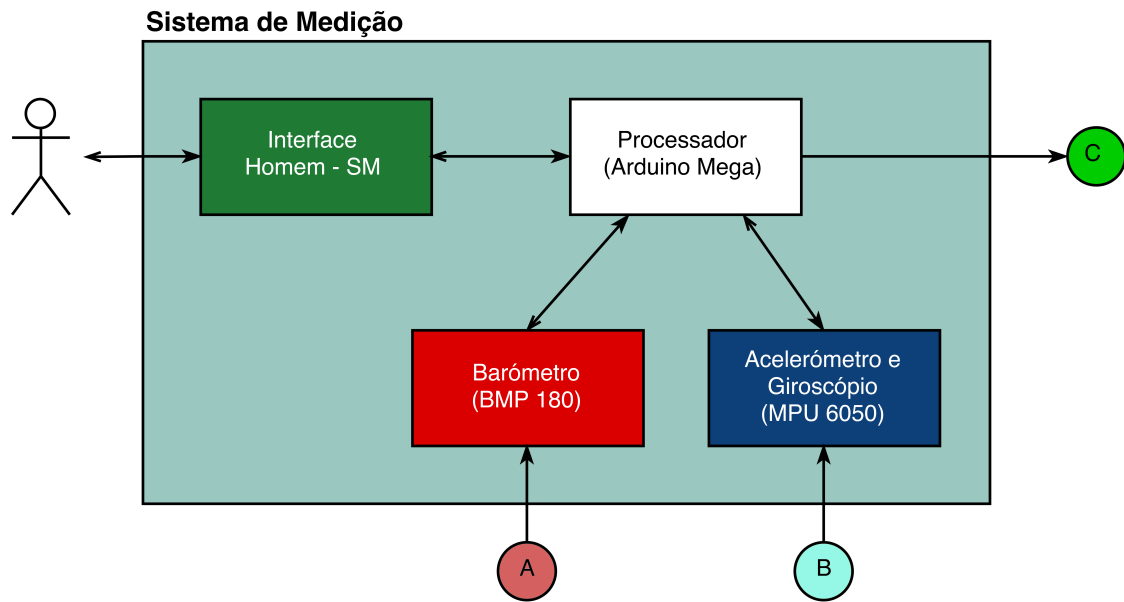


Figura 3.8: Arquitetura de alto nível do SM

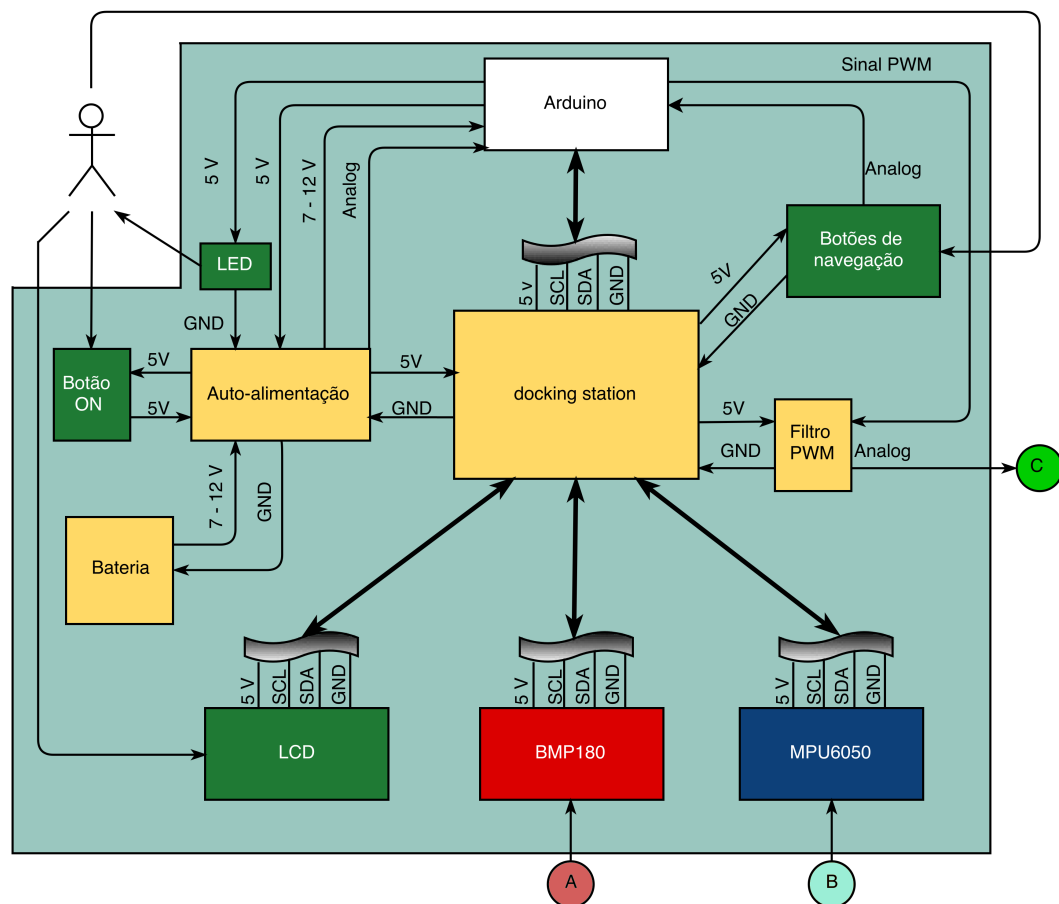


Figura 3.9: Arquitetura de baixo nível do SM

Esta arquitetura de baixo nível não discrimina todos os componentes e ligações do SM. Alguns dos componentes são representados por blocos. Optou-se por apresentar deste modo por questões de organização espacial da página e do documento. A discriminação destes blocos são apresentados mais adiante.

3.5.3 Especificações Tecnológicas

As especificações tecnológicas do SM são as seguintes:

- A alimentação do SM é feita por uma bateria de 9V. Contudo, o circuito está preparado para receber outro tipo alimentação desde que não exceda os 12 V e tenha pelo menos 250 mAh;
- É utilizado um relé (V23100 V4005-A000) para efetuar a auto alimentação do circuito, permitindo desligar o SM recorrendo a software;
- A interface SM-Homem é realizada por LCD (16x2), um LED vermelho e cinco botões de impulso (quatro para navegação e um para ligar);
- A medição de vibrações e acelerações é conseguida pela utilização do MPU6050. O BMP180 mede a pressão atmosférica;
- O processamento dos dados é feito por um Arduino Mega 2560;
- É incorporado um *Wierless SD Shield* para expandir as capacidades do Arduino, permitindo utilizar um cartão SD para guardar a informação gerada;
- A comunicação entre Arduino, sensores e LCD é feita utilizando o protocolo I²C;
- Na *cable dooking station* são agrupados os cabos utilizados no protocolo I²C (Vcc, GND, SCL e SDA);
- O envio de informação sob forma analógica é feito recorrendo a um filtro passa baixo. Este converte o sinal digital PWM do Arduino em sinal analógico;
- O hardware encontram-se encapsulados numa caixa com as dimensões (255 × 190 × 90) mm.

3.5.4 Especificações Operacionais

As especificações operacionais do SM são as seguintes:

- O SM é ligado premindo o botão de impulso até o LED vermelho acender;
- No ecrã do LCD é apresentado um menu no qual o utilizador pode navegar recorrendo a botões de impulso. No canto superior direito do ecrã do LCD existe um ícone que indica o nível de bateria (Figura 3.10(b));
- A primeira ação a realizar após ligar o SM é escolher no menu, qual a operação que pretende efetuar (Figura 3.10(a));
- Quando selecionada a opção, caso exista necessidade, é feito o zero aos sensores. Após concluída a ação anterior, são iniciadas as medições;
- O SM recolherá dados, até o utilizador carregar num botão para parar, ou caso a bateria esteja quase esgotada;
- A informação recolhida e gerada no SM é guardada num cartão de memória de SD de 16Gb com a extensão **.txt*. O nome do ficheiro criado, depende da opção escolhida no menu. O texto gerado tem uma formatação que facilita a importação de dados para programas como, por exemplo, Excel ou Matlab.



(a) Esquema do sistema de menus completo



(b) Sistema de menus num LCD 16x2

Figura 3.10: Sistema de menus utilizado no SM

3.5.5 Hardware

A Figura 3.11 é uma fotografia do hardware utilizado no SM e a Tabela 3.4 é a lista dos componentes observados na figura.

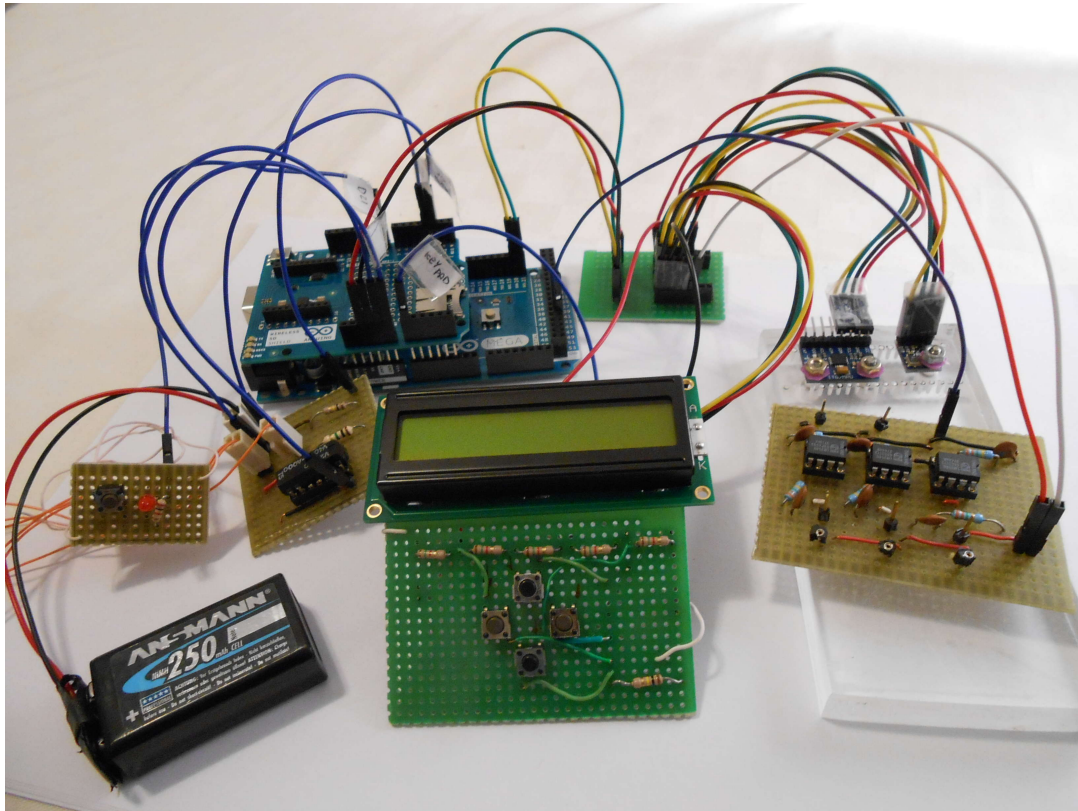


Figura 3.11: Hardware do SM

Componente	Quantidade	Componente	Quantidade
Arduino Mega	1	Relé V23100 V4005-A000	1
AmpOp LM358-N	3	Resistor 1 K Ω	1
BMP180	1	Resistor 1.5 K Ω	5
Botão de impulso	5	Resistor 2 K Ω	1
Condensador 1 nF	6	Resistor 100 K Ω	1
Díodo D1N4148	1	Resistor 3.2 M Ω	6
LCD 16x2	1	Scoket de 4 pinos	3
LED vermelho	1	Socket de 7 pinos	1
MPU6050	1	Wireless SD Shield	1

Tabela 3.4: Lista de componentes

Arduino

O Arduino (Figura 3.12) é uma plataforma, de código aberto, destinada à prototipagem. As vantagens desta tecnologia são o seu preço, flexibilidade e fácil utilização. Uma outra vantagem, não menos importante, é a existência de uma grande comunidade que utiliza esta tecnologia, que partilha na Web ideias, dúvidas, sugestões, código e projetos. Esta comunidade conta com estudantes, investigadores, artistas, entusiastas, profissionais e muitos outros que disponibilizam gratuitamente os seus projetos e ajudam os novos membros [33].

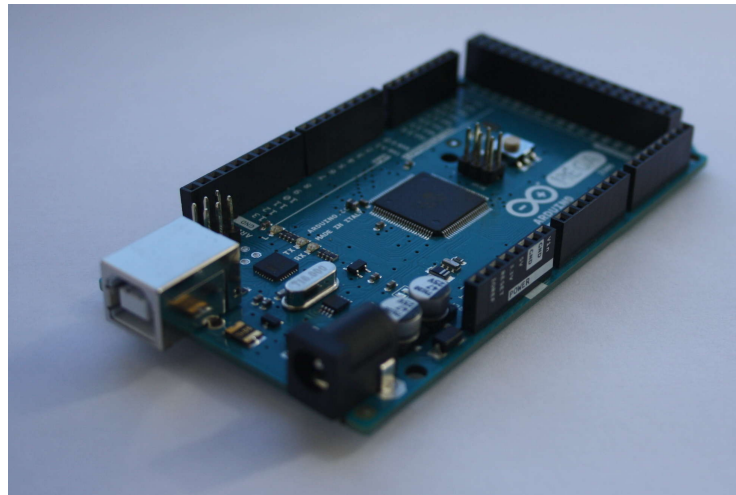


Figura 3.12: Placa Arduino Mega

A placa Arduino utilizada no SM é a Arduino Mega 2560 e as suas especificações são as seguintes:

Microcontroller	ATmega1280
Operating Voltage	5V
Input Voltage	7-12V
Digital I/O Pins	54
Analog Input Pins	16
DC Current per I/O Pin	40 mA
Flash Memory	128 KB
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Tabela 3.5: Especificações técnicas do Arduino MEGA [33]

Sensores

As especificações tecnológicas dos sensores foram apresentadas na secção 3.2. Os sensores foram colocados numa placa de acrílico com 12 mm de espessura com o intuito de manter o acelerómetro o mais nivelado possível. Verifica-se que na Figura 3.13 foram colocadas chapas de aço sobre o acrílico para ajudar a fixar o SM ao chão da cabina, tal como sugere a norma ISO 18738.

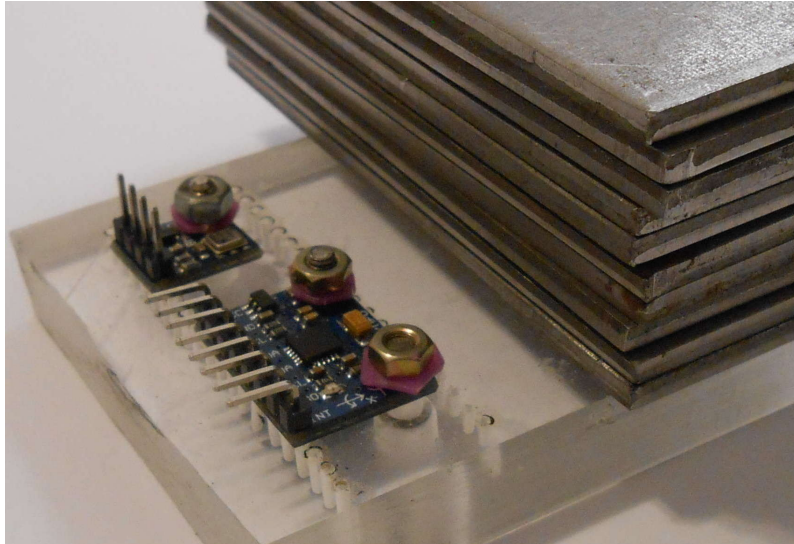


Figura 3.13: Placa de acrílico com sensores e chapas de aço

Cable Docking Station

A esta placa (Figura 3.14) foi atribuído o nome de *Cable Docking Station* devido à função pela qual foi concebida: agrupar o maior número de cabos de modo a que as ligações internas do hardware do SM sejam mais facilmente percebidas, reparadas ou modificadas. A Docking Station permite ainda que os componentes tenham alguma liberdade para que sejam fixados em locais mais convenientes no momento em que se processa o encapsulamento do SM.

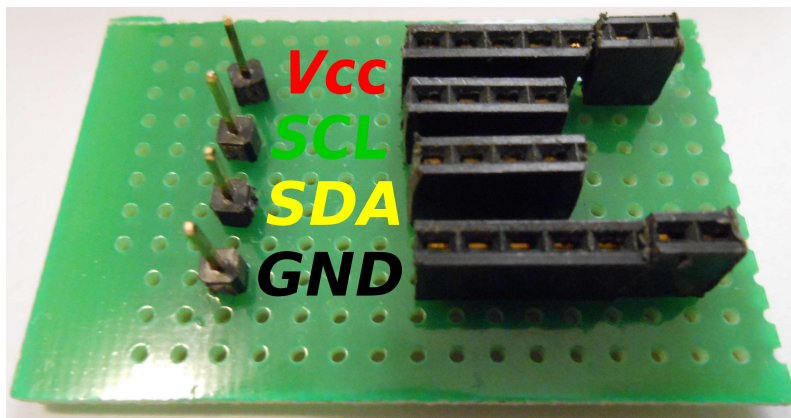


Figura 3.14: Cable docking station

Os cabos são ligados num dos *Headers* consoante seja Vcc, SCL, SDA ou GND. Os *Male Headers* servem para estabelecer a ligação Arduino-Docking Station. Os *Female Headers* são utilizados pelos componentes que necessitem de Vcc, SCL, SDA e/ou GND.

Circuito de Alimentação

O Arduino não possui um botão ON/OFF, ou seja, existe sempre consumo de energia. Para resolver este problema foi desenvolvida uma placa à qual foi dada o nome de *Circuito de Alimentação* (Figura 3.15). Esta está dividida em duas placas mais pequenas para permitir que o circuito principal seja colocado próximo da bateria, enquanto que o conjunto LED-botão é colocado numa das faces do encapsulamento, visto que o último é utilizado como interface pelo utilizador.

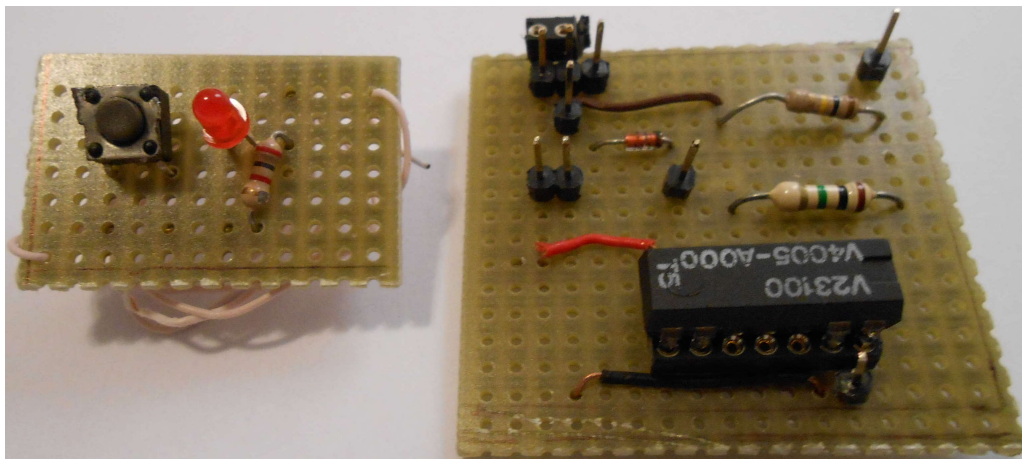


Figura 3.15: Circuito de alimentação

Segundo o website oficial da Arduino é recomendável alimentar a placa pela porta VIN com uma tensão de 7-12 Volt. Considera-se extremamente desaconselhável alimentá-la nas portas "3.3V" ou "5V", porque pode danificar o Arduino [33]. Este necessita efetivamente de 5 Volt para funcionar, contudo, a tensão de alimentação sugerida é de 7-12, porque existe um regulador de tensão embutido no Arduino.

A alimentação do SM é feita com recurso a tecnologia de relés. O esquema eletrónico é representado na Figura 3.16. Na Figura 3.16, a sigla A10 representa a entrada analógica 10 do Arduino. Nesta, é lida a tensão da bateria com recurso a um divisor de tensão. Caso a bateria esteja quase esgotada, o SM informa o utilizador da necessidade da sua substituição, guardando os dados recolhidos (caso existam), desligando-se.

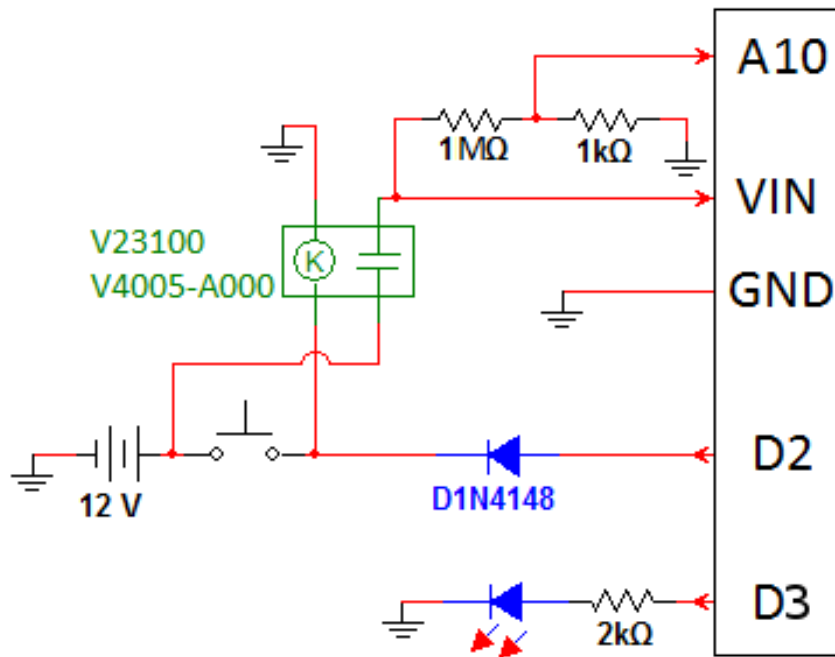


Figura 3.16: Circuito de alimentação, esquema eletrônico

Esta possibilidade de o SM se desligar consegue-se utilizando um circuito de auto alimentação. Os componentes deste circuito são um relé, um botão de impulso, um LED e um diodo. O funcionamento da auto alimentação é o seguinte:

- Pressiona-se o botão de impulso;
- O relé é ativado e é iniciada a passagem de corrente da bateria para o Arduino;
- O Arduino demora alguns milésimos de segundo a inicializar-se. Durante este tempo o botão de impulso não pode ser largado;
- O Arduino inicializa-se e, pela porta digital D2, alimenta o relé. A porta digital D3 acende o LED vermelho. Nesta altura o botão pode ser libertado.

Por fim, para desligar o SM o utilizador escolhe a opção “Desligar” apresentada no LCD. Esta opção desativa a porta digital D2 que corta a auto alimentação.

Interface SM-Homem

A interface SM-Homem é constituída por um LCD, quatro botões de navegação (Figura 3.17), o LED e o botão apresentados previamente.

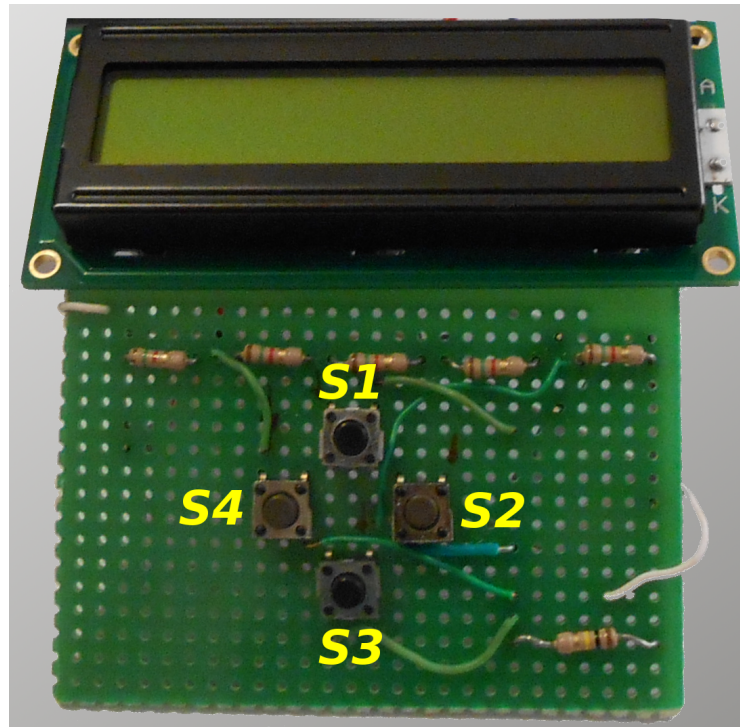


Figura 3.17: Interface SM-Homem

No LCD é apresentado um menu no qual o utilizador pode navegar e seleccionar uma opção. A navegação é feita por quatro botões de impulso organizados em forma de cruz. Os botões verticais (S1 e S3) são utilizados para navegação. O botão S2 é utilizado para seleção e o botão S4 para voltar ao menu anterior. Os botões foram desenhados para funcionarem em modo analógico (Figura 3.18). Este modo de funcionamento, relativamente ao digital, tem as vantagens de: utilizar apenas uma entrada do Arduino e associar a cada botão uma gama de tensão.

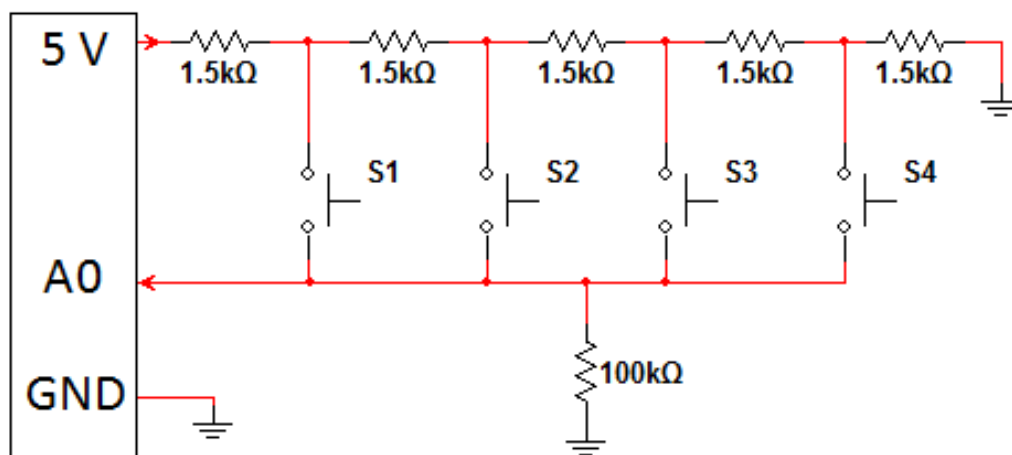


Figura 3.18: Circuito para navegação no menu

Filtro PWM

O Arduino pode ler sinais analógicos e digitais, mas apenas consegue enviar os últimos. O sinal digital PWM (*Pulse-width modulation*) é facilmente convertido em analógico quando utilizado um filtro passa baixo. O filtro utilizado é de segunda ordem, com o objetivo de obter um resultado com mais atenuação. A Figura 3.19 representa o esquema eletrônico proposto para o filtro. Recorrendo à Equação (3.11) é possível escolher os condensadores e aos resistores em função da frequência de corte (f_c) do filtro. Para o correto funcionamento do filtro, a frequência de corte escolhida deve ser uma década inferior à frequência do sinal PWM. O AmpOp LM358-N é instalado para fazer desacoplamento de impedâncias na entrada e saída do filtro.

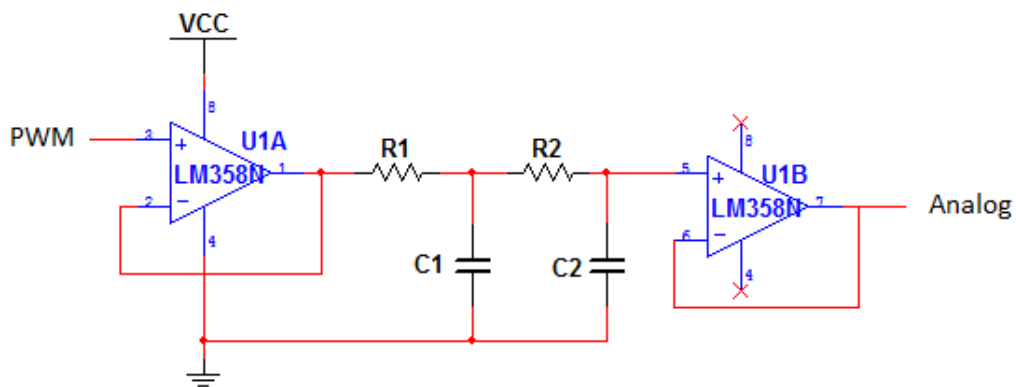


Figura 3.19: Esquema eletrônico para filtro PWM

$$f_c = \frac{1}{2\pi\sqrt{R_1 \cdot C_1 \cdot R_2 \cdot C_2}} \quad (3.11)$$

O Arduino Mega 2560 oferece saídas PWM nas portas 2 a 13. Por definição do fabricante, nas portas digitais 4 e 13 a frequência do sinal PWM gerado é de aproximadamente ³ 1000 Hz e nas restantes portas é de 500 Hz.

O fabricante oferece a possibilidade de alterar a frequência do sinal PWM. Contudo, esta configuração altera o relógio interno do Arduino, fazendo com que 1000 milissegundos deixem de corresponder a 1 segundo.

Na Figura 3.20 observa-se a placa intitulada *Filtros PWM*. Nesta foram dimensionados três filtros com frequência de corte de 50 Hz. Com estes três filtros é possível enviar três sinais. Os sinais PWM gerados no Arduino têm a resolução de 8 bits.

³ O valor exato é 976.5625 Hz para os primeiros e 490.20 Hz para os segundos.

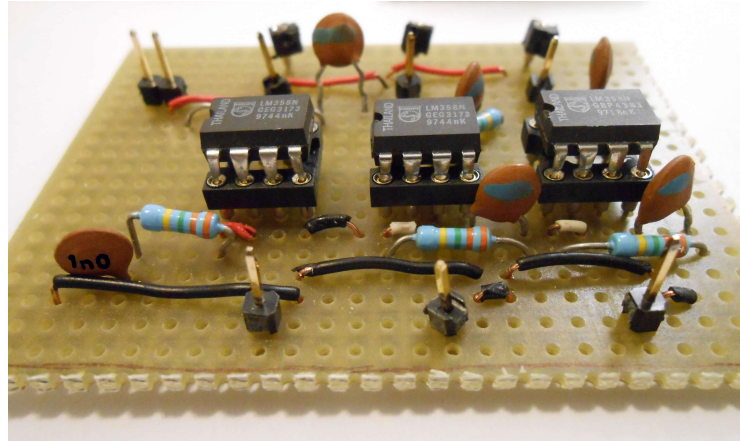


Figura 3.20: Placa com três filtros passa baixo

3.6 Descrição Sumária do Elevador Protótipo

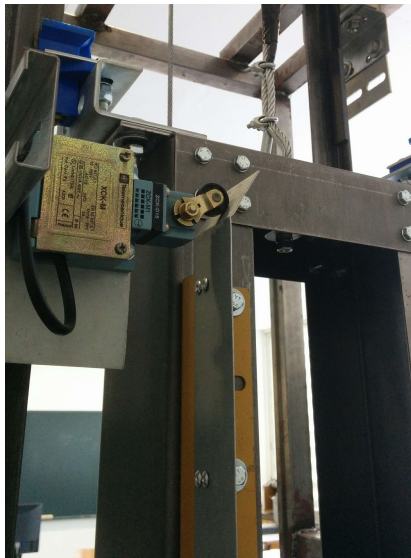
A Figura 3.21 é uma fotografia do protótipo de elevador monta-cargas, desenvolvido no trabalho de Mestrado do aluno Nuno Ângelo. Para contextualizar o leitor com o objeto em estudo (o elevador modelo) são apresentadas as suas especificações de forma sumária. Para mais detalhes sobre o modelo de elevador estarão disponíveis aquando da publicação da tese de Nuno Ângelo intitulada *Projeto de Sistema de Automação de Montacargas Industrial*.



Figura 3.21: Elevador modelo

3.6.1 O Elevador Monta-Cargas

O elevador (Figura 3.21) serve três patamares e está montado numa estrutura com $(1.08 \times 0.88 \times 2.30)$ metros. No lado esquerdo da estrutura (Figura 3.21) está representada a caixa do elevador. Nesta, encontram-se a cabina, o contrapeso, dois sensores mecânicos de fim-de-curso (superior e inferior), quatro sensores magnéticos de proximidade e duas guias. Os fins-de-curso não atuam na cabina, mas sim no contrapeso. Deste modo, o fim-de-curso inferior é o que se encontra mais próximo do patamar 2 (Figura 3.22(a)). As guias têm ímanes colocados em posições estratégicas (Figura 3.22(b)). No topo da cabina encontram-se quatro sensores (Figura 3.23) que funcionam como sensores de posicionamento magnético. Na Figura 3.23 os sensores magnéticos pintados a verde são utilizados quando a cabine sobe e os pintados a azul são utilizadas na descida.



(a) Pormenor de contrapeso a atuar no fim-de-curso inferior



(b) Guia esquerda com ímanes identificando a laranja

Figura 3.22: Pormenores da caixa do elevador modelo

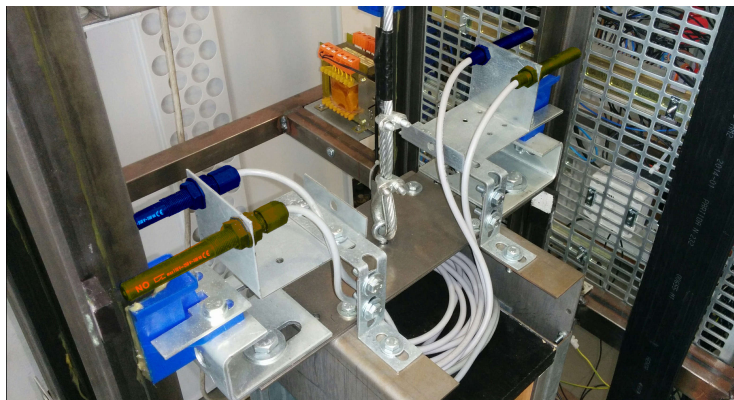


Figura 3.23: Pormenor das ampolas magnéticas no topo da cabina

A casa das máquinas corresponde ao topo e ao lado direito da estrutura (Figura 3.21).

Nela encontram-se a máquina de tração, o PLC, o variador de velocidade, a botoneira e alguns atuadores e proteções elétricas. A máquina de tração (Figura 3.24(a)) é um motor trifásico cuja função é imprimir rotação a uma roda de aderência, sendo a rotação definida pelo PLC e imposta pelo variador de velocidade. A Figura 3.24(b) mostra a tecnologia utilizada neste elevador. A botoneira (Figura 3.24(c)) foi colocada dentro da casa das máquinas para facilitar a sua utilização. Na situação real encontrar-se-ia dentro da cabina. Na botoneira observa-se um botão de paragem de emergência e três botões para programar o movimento da cabina.

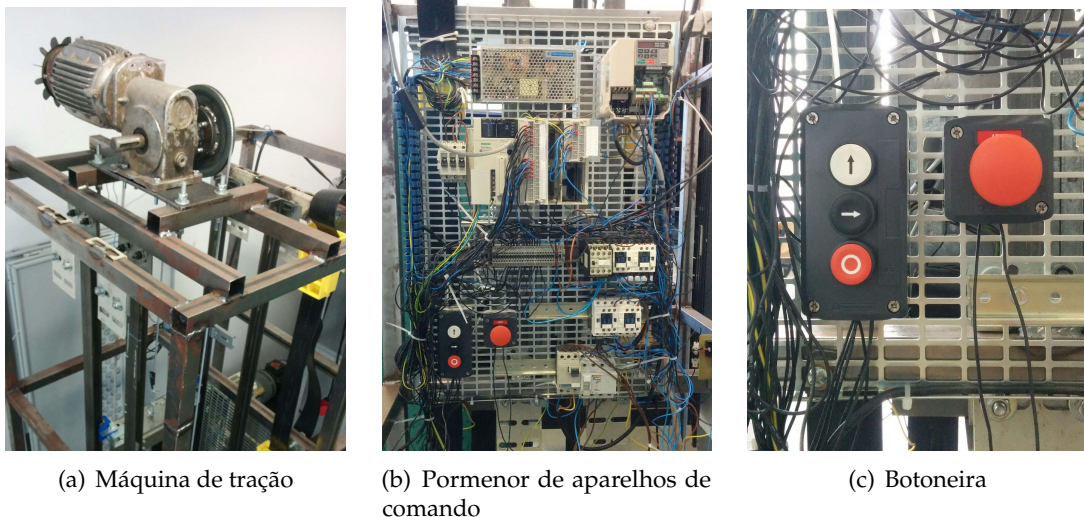


Figura 3.24: Pormenores da casa das máquinas do elevador modelo

3.6.2 Especificações Funcionais, Tecnológicas e Operacionais

Especificações Funcionais

O elevador tem dois modos de funcionamento: um automático e outro manual.

Em modo automático a interação Homem-Máquina é realizada pela botoneira representada na Figura 3.24(c). No caso do modo manual (apenas disponível para pessoal técnico) a interação Homem-Máquina é feita pela consola HMI (Figura 3.25) presente no variador de velocidade. Contudo, o modo manual não deve ser utilizado, uma vez que obriga o motor a trabalhar em esforço.

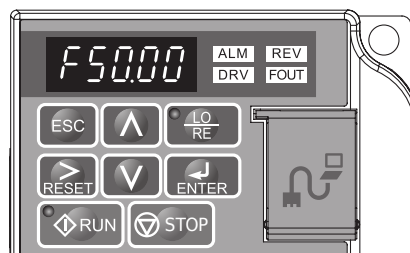


Figura 3.25: Consola HMI do variador de velocidade [34]

Especificações Tecnológicas

Utiliza-se uma tecnologia de relés, variador de velocidade (Omron-V1000) e PLC (TSX Micro 3721). O variador tem implementado um controlador PI que utiliza como referência o valor enviado pelo PLC.

Os sensores mecânicos de fim-de-curso são ativados, caso a cabina exceda os patamares 0 ou 2. Os sensores de posicionamento magnéticos informam se a cabina se encontra entre patamares ou no patamar.

Especificações Operacionais

A primeira ação a realizar quando se liga o elevador é premir o botão vermelho (paragem de emergência). Em seguida, pressiona-se botão (0) ou (↑) para acertar a cabina ao primeiro ou a último patamar. Após a chegada da cabina ao patamar, o elevador está pronto a funcionar. As opções de funcionamento são as seguintes:

- Premir o botão (0) para a cabina se mover para o patamar 0.
- Premir o botão (→) para a cabina se mover para o patamar 1.
- Premir o botão (↑) para a cabina se mover para o patamar 2.
- Premir o botão de paragem de emergência para efetuar a paragem imediata do sistema e reinicializar as memórias do PLC.

Notas adicionais:

Sempre que seja acionada a paragem de emergência, é necessário fazer o acerto ao patamar 0 ou 2.

Só após a chegada ao patamar é possível executar as ações dos botões (0), (→) ou (↑).

Se por acaso o contrapeso acionar um dos fins-de-curso, a máquina de tração é forçada a parar. Após três segundos, a cabina move-se para o patamar mais próximo (0 ou 2).

3.6.3 Arquitetura de Alto Nível

A figura 3.26 tenta representar de forma simples o funcionamento do modelo do elevador, ou seja representa os componentes e respetivas interações. De forma muito genérica o funcionamento é o seguinte: o utilizador, recorrendo à botoneira, escolhe o patamar de destino da cabina. O PLC interpreta os comandos, ativa os atuadores necessários e envia um valor de referência para o variador. A corrente sai do variador, passa pelos atuadores e aciona a máquina de tração. A máquina de tração por sua vez faz movimentar a cabina. Durante o seu movimento os sensores estão constantemente a ser lidos e a sua informação processada pelo PLC.

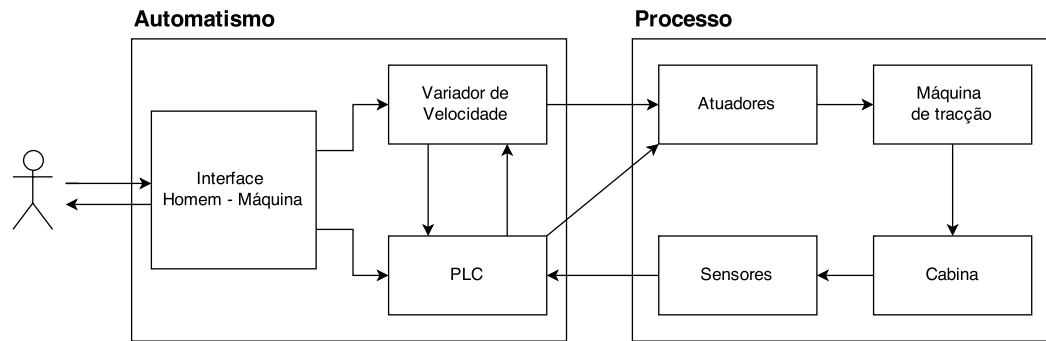


Figura 3.26: Arquitetura de alto nível do modelo de elevador

3.6.4 Arquitetura de Controle

O controlo do elevador consiste em monitorizar a frequência aplicada na máquina de tração de modo a que a cabina tenha o comportamento desejado. A Figura 3.27 representa a arquitetura de controlo. Nesta observa-se que a um determinado valor de tensão corresponde uma frequência que impõe uma velocidade linear na cabina.

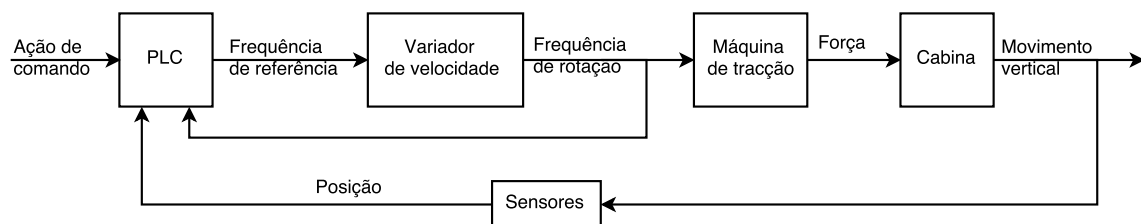


Figura 3.27: Arquitetura de controlo do elevador modelo

Em regime estacionário pode-se recorrer à Equação (3.12) para efetuar a conversão entre a frequência imposta à máquina de tração, a tensão de referência enviada pelo PLC e estimar a velocidade linear da cabina.

$$\text{Frequência [Hz]} = \frac{10}{2} \times \text{Tensão [V]} = 177.482 \times \text{Velocidade [m/s]} - 0.314 \quad (3.12)$$

A ação de controlo indicada na Figura 3.27 consiste em definir o patamar de destino da cabina. Para o PLC saber qual o patamar em que se encontra a cabina são lidos e interpretados endereços (%I1.x). Para colocar a cabina num patamar é necessário garantir que %I1.6 = 1 e %I1.7 = 1 e que um dos os requisitos apresentados na Tabela 3.6 seja também verificado.

Patamar 0	<i>subida</i> para patamar 1	<i>descida</i> para patamar 1	Patamar 2
%I1.8 = 0	2x transição 0/1 de %I1.6	2x transição 0/1 de %I1.7	%I1.9 = 0

Tabela 3.6: Condições para identificação o patamar

Os endereços aqui utilizados encontram-se discriminados na Tabela 3.7.

3.6.5 Lista de Pontos

O PLC utilizado é o TSX Micro 3721 da Schneider e tem instalado o módulo DMZ28DR (entrada-saída digital) na posição #1 e o módulo AMZ600 (entrada-saída analógica) na posição #5. Na Tabela 3.7 apresentam-se todas as entradas e saídas disponíveis.

Descrição	Endereço	Sinais				Obs.
		ED	EA	SD	SA	
Botão (0)	I 1.0	x				NA
Botão (→)	I 1.1	x				NA
Botão (↑)	I 1.2	x				NA
Paragem de emergência	I 1.3	x				NF
Fim-de-curso superior	I 1.4	x				NF
Fim-de-curso inferior	I 1.5	x				NF
Ampola de descida	I 1.6	x				NA
Ampola de subida	I 1.7	x				NA
Ampola biestável de descia	I 1.8	x				NA
Ampola biestável de subida	I 1.9	x				NA
Frequência à saída do variador	IW 5.0		x			[0; 10000]
Descida da cabina	Q 2.0			x		NA
Subida da cabina	Q 2.1			x		NA
Freio da máquina de tração 1	Q 2.2			x		NA
Freio da máquina de tração 2	Q 2.3			x		NA
Sinal de controlo do PLC	QW 5.4				x	[0; 10000]
Não utilizados	I 1.10 – I 1.15	x				
	IW 5.1 – IW 5.3		x			
	Q 2.4 – Q 2.11			x		
	QW 5.5				x	

Tabela 3.7: Lista de pontos do TSX 3721

3.7 Modelação do Comportamento do Variador de Velocidade do Elevador Modelo

Para determinar o modelo que representa o elevador protótipo desenvolveu-se um programa que consiste num ciclo no qual a cabina é forçada a deslocar-se entre fins-de-curso. A sua velocidade (lenta, normal ou rápida) foi definida pelos botões (0), (→) ou (↑) respetivamente. Os botões foram pressionados aleatoriamente e, recorrendo a uma placa de aquisição de dados, obteve-se a Figura 3.28.

Na Figura 3.28 a vermelho está represento o sinal de entrada (valor de referência enviado pelo PLC) e a azul o sinal de saída (variador de velocidade). Entre os 20 e 25 segundos observa-se uma variação brusca do sinal azul. Tal facto, deve-se à chegada da cabina ao fim-de-curso e na imediata mudança de direção.

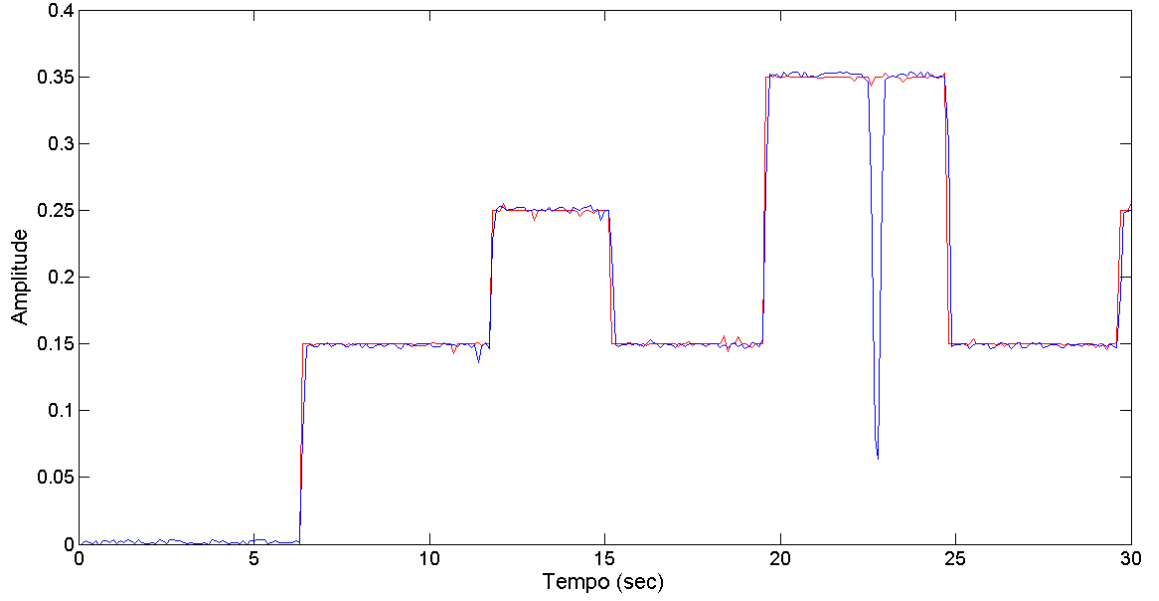


Figura 3.28: Variação da velocidade ao longo do tempo

O melhor resultado obtido foi um modelo ARX(2,2,1) representado pela Equação (3.13).

$$y(k) - 0.9532 y(k-1) + 0.3255 y(k-2) = 0.5751 u(k-1) - 0.2044 u(k-2) + e(k) \quad (3.13)$$

A Figura 3.29 representa a resposta do modelo ao degrau. Para obter resultados aceitáveis em simulação são necessárias pelo menos cinco amostras antes de $t = \tau$, neste instante o sinal de saída corresponde a 63% do resultado final. Com este número de amostras, obtido empiricamente, a simulação da resposta do sistema em regime transitório é melhor. A frequência de aquisição foi 10 Hz e, sendo imposta pelo PLC, não é possível aumentá-la. Para contornar esta dificuldade, e obter melhores resultados em simulação, foram realizadas dez modelações consoante a tensão aplicada no variador. Os modelos ARX(2,2,1) obtidos são representados pela Equação (3.14). As constantes a_1, a_2, b_1 e b_2 são apresentadas na Tabela 3.8.

$$y(k) = -a_1 \cdot y(k-1) - a_2 \cdot y(k-2) + b_1 \cdot u(k-1) + b_2 \cdot u(k-2) + e(k) \quad (3.14)$$

Em simulação foi construído um multi-modelo que, conforme o valor de tensão, utiliza um ou uma combinação de dois modelos ARX. Por exemplo se a tensão for 1.25 Volt o resultado de simulação utiliza 50% do modelo 1 Volt e 50% do modelo 1.5 Volt.

Nas Figuras 3.30 e 3.31 são representados três sinais. O sinal preto representa a referência, os sinais azul e laranja são a resposta do controlador PI ao sinal preto. O sinal laranja é o resultado de simulação e o azul é o sinal no elevador.

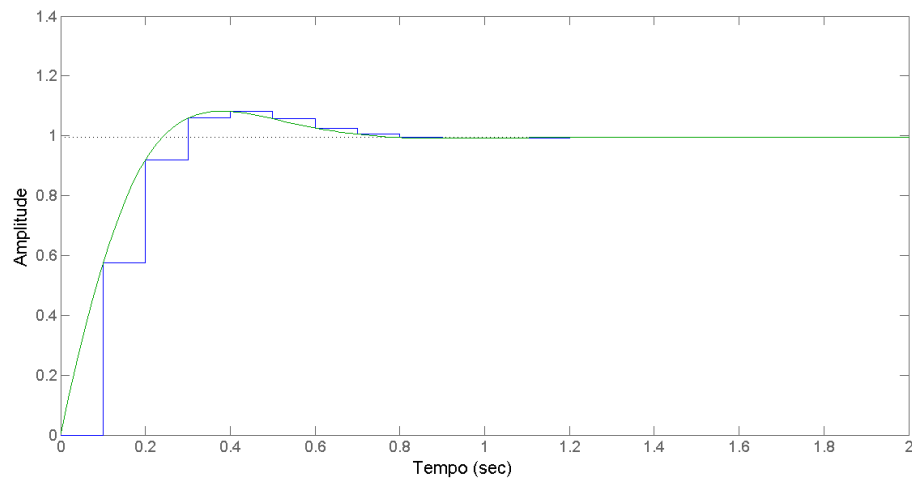


Figura 3.29: Resposta do modelo ao degrau unitário

Tensão (Volt)	a_1	a_2	b_1	b_2
1.0	-0.6894	0.09238	0.7351	-0.3171
1.5	-0.7323	0.1874	0.5724	-0.1065
2.0	-0.5591	0.2023	0.3901	0.2618
2.5	-0.1732	0.02285	0.3929	0.4777
3.0	-0.5629	0.1619	0.5446	0.06451
3.5	-0.364	0.09249	0.4039	0.3373
4.0	-0.5091	0.1324	0.2639	0.3679
4.5	-0.6733	0.2439	0.3092	0.2710
5.0	-0.8748	0.2793	0.159	0.2492
5.5	-0.9049	0.3213	0.2256	0.192

Tabela 3.8: Parâmetros do modelo ARX(2,2,1) em função da tensão aplicada no variador de velocidade

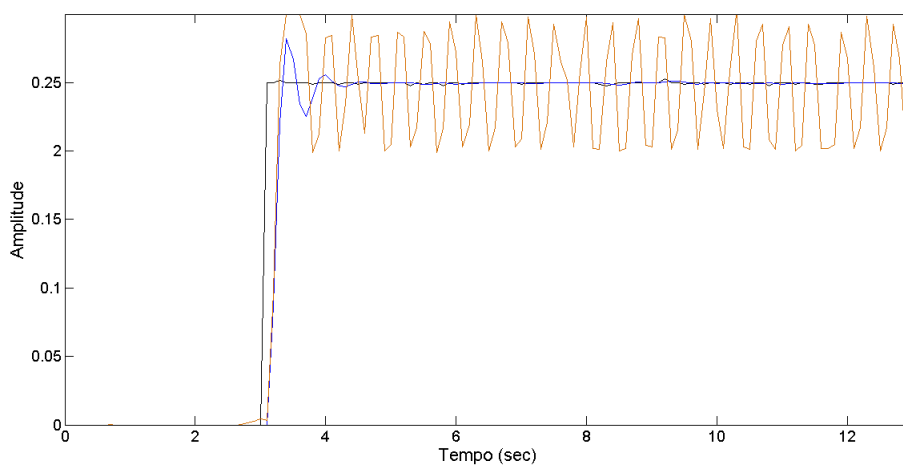


Figura 3.30: Resposta do modelo ao degrau

Após afinar o controlador no elevador modelo, os parâmetros do controlador foram utilizados em simulação e o resultado concretiza-se na Figura 3.31.

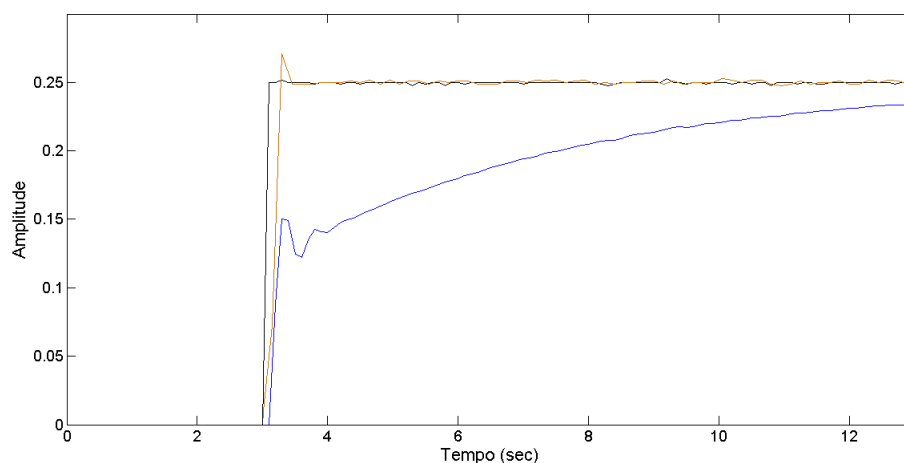


Figura 3.31: Resposta do modelo ao degrau

Verifica-se que a modelação linear do elevador não é eficaz para projetar controladores, porque o elevador tem dinâmica não-linear. Deste modo os controladores foram sintonizados no elevador.

3.8 Controlo do Elevador Modelo

O PLC utilizado tem a possibilidade de ser programado em qualquer uma das seguintes linguagens: *Ladder Diagram*, *Instruction List*, *Structured Text* e *Sequential Function Charts*. O PLC foi programado em Structured Text (ST) uma vez que esta linguagem é normalmente utilizada em sistemas complexos.

3.8.1 Estrutura do Código

O código foi estruturado para ser facilmente reaproveitado e alterado, uma vez que os programas correm em multitarefa. A Figura 3.32 é uma representação gráfica do código para uma fácil e rápida compreensão. É facultado em apêndice o código escrito em ST (Apêndices A, B e C).

Na Figura 3.32 estão representadas três das tarefas (*tasks*) criadas para o funcionamento do elevador, identificadas com números de 1, 2 e 3.

Tarefa 1 - Interpretar – verifica duas situações: se são ativados os fins-de-curso ou se o utilizador deseja enviar a cabina para um patamar. Caso uma destas situações ocorra a Tarefa 2 é ativada;

Tarefa 2 - Gerador – identifica, se possível, o patamar onde se encontra a cabina. Se existir informação sobre o patamar no qual a cabina se encontra,

então é enviada informação sobre o sentido do movimento da cabina. Caso contrário, a informação enviada será para a cabina descer. Esta informação é enviada para a Tarefa 3;

Tarefa 3 - Mover – ativa um perfil de velocidades (Tarefa 4) e verifica se a cabina chegou ao patamar de destino;

Tarefa 4 – perfil de velocidades que se pretende utilizar.

As Tarefas 2, 3 e 4 só são ativadas quando ocorrem determinadas condições. São desativadas quando essas condições não se verificam. Pelo que, apenas a Tarefa 1 é que é executada em todos os ciclos do autómato. Por este facto, a ação de paragem de emergência do sistema encontra-se na Tarefa 1. Quando se ativa esta ação, o sistema pára imediatamente e as memórias são reinicializadas.

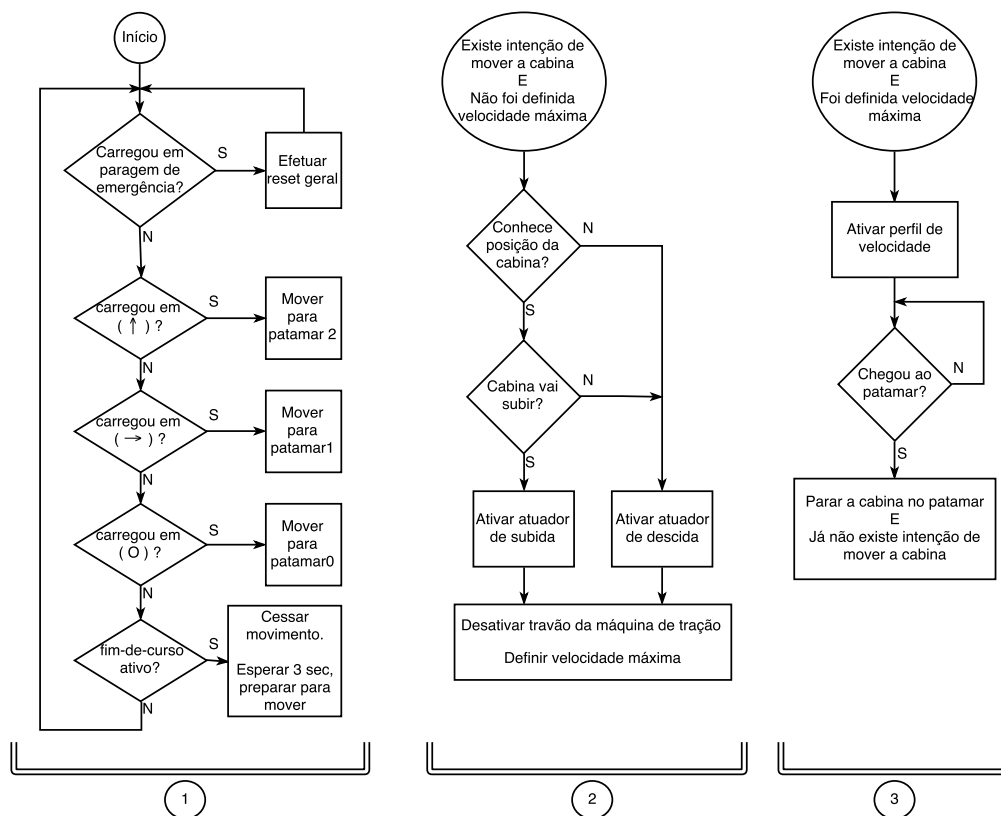


Figura 3.32: Representação gráfica das Tarefas 1, 2 e 3

3.8.2 Perfis de Velocidade

Para avaliar o conforto no modelo de elevador foram desenhados seis perfis de velocidade, baseadas naqueles que foram apresentados no Capítulo 2. Nos quatro primeiros casos, a posição da cabina é conhecida apenas em alguns pontos do percurso, nos restantes dois, a posição é sempre conhecida. Para todos os casos, os perfis de velocidades são apresentados sob a forma de esquemas lógicos do automatismo (logigramas).

Nos logigramas, a letra V representa o valor de velocidade pretendida para desenhar o perfil. Para fornecer a possibilidade de alterar a dinâmica do processo, no código ST, as duas últimas linhas são uma das seguintes opções:

<pre>1 (* SET %M3;*) (*com controlador*) 2 %QW5.4:=%MW17; (*sem controlador*)</pre>	<pre>1 SET %M3; (*com controlador*) 2 (* %QW5.4:=%MW17;*) (*sem controlador*)</pre>
---	---

Se a linha de código que tem $SET \%M3$ estiver comentada (situação da esquerda), então a dinâmica do sistema mantém-se. Se não estiver comentada (situação da direita) então é ativado um controlador Proporcional e Integral (PI) que altera a dinâmica do sistema (Figura 3.33) e a implementação em ST está no Apêndice C.

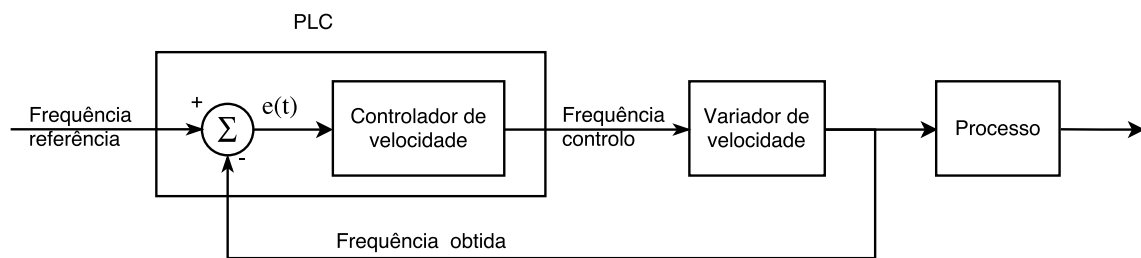


Figura 3.33: Diagrama de blocos quando ativado o controlador PI

Perfil Básico

Dos seis perfis de velocidade referidos, este é o mais básico (Figura 3.34). Para este caso, o logigrama (Figura 3.35) tem de garantir que o elevador (em regime estacionário) ou está parado ou está com velocidade. A primeira decisão a efetuar é determinar a velocidade da cabina, ou seja: a velocidade da cabina deve ser a adequada à distância, tal como foi apresentado nas Tabelas 2.6 e 2.7.

O modelo tem três patamares. Consoante se desloque um ou dois patamares é definida uma velocidade diferente, respectivamente V_1 ou V_2 ($V_2 > V_1$). O código ST para este perfil está em: Apêndice A.

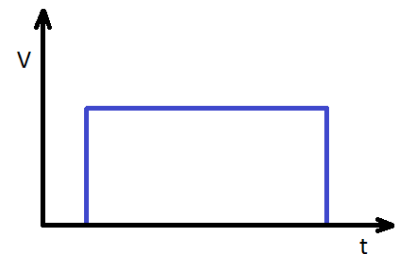


Figura 3.34: Perfil básico

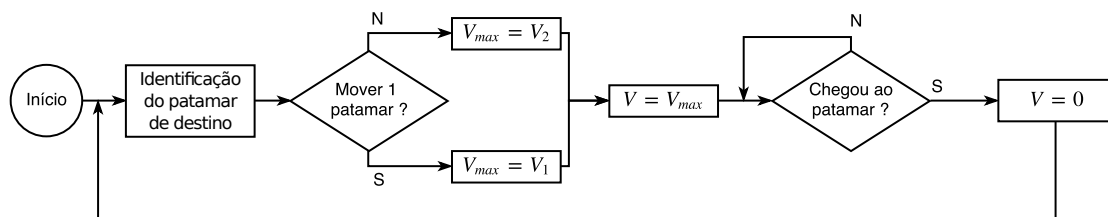


Figura 3.35: Logigrama para produzir o perfil básico

Perfil Básico com Velocidade de Aproximação

A diferença deste perfil (Figura 3.36) relativamente ao anterior é que próximo do patamar a cabina passa a deslocar-se com uma velocidade de aproximação V_3 ($V_2 > V_1 > V_3$), garantindo uma paragem mais precisa. Como é apresentado pela Figura 3.37, existe apenas a necessidade de tomar atenção ao sensor que indica que o patamar de destino está próximo.

O código ST para este perfil está em: Apêndice A.

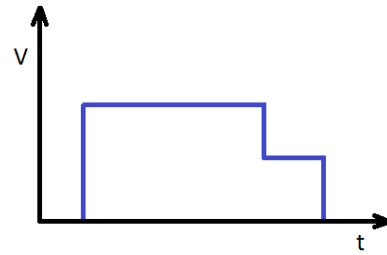


Figura 3.36: Perfil básico com velocidade de aproximação

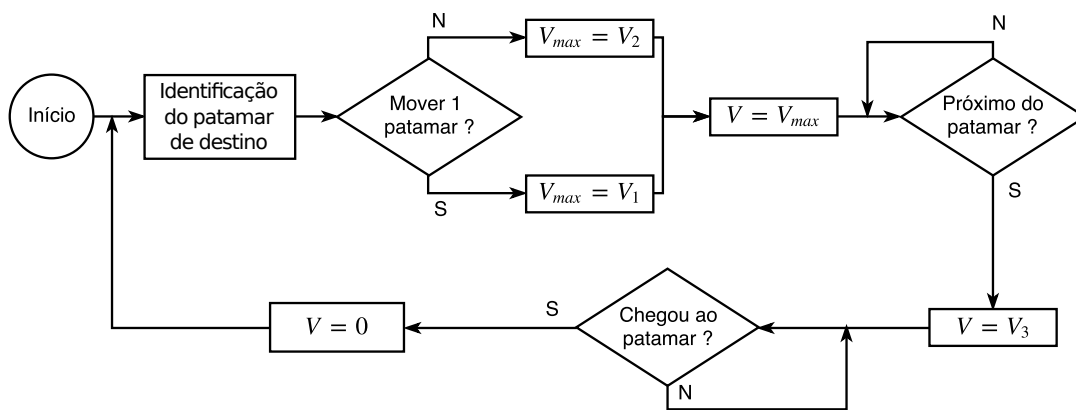


Figura 3.37: Logigrama para produzir o perfil básico com velocidade de aproximação

Perfil em Rampa

O perfil da Figura 3.38 representa o perfil da situação real pouco utilizada. A velocidade aumenta gradualmente até um valor máximo, mantendo-se durante a maior parte do percurso com essa velocidade. Por fim, reduz gradualmente até parar no patamar. No logigrama (Figura 3.39) a velocidade V_4 representa o incremento da velocidade, enquanto que a velocidade V_5 representa a velocidade mínima que garante a rotação da máquina de tração.

O código ST para este perfil está em: Apêndice A.

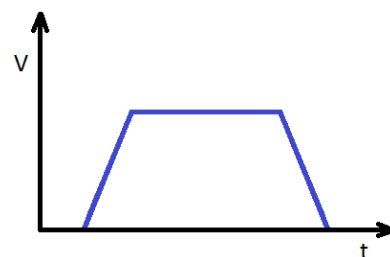


Figura 3.38: Perfil em rampa

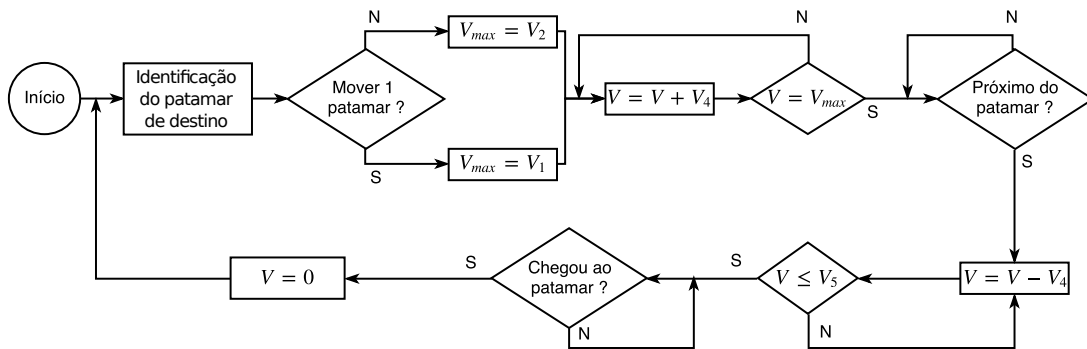


Figura 3.39: Logigramma para produzir o perfil em rampa

Perfil em Rampa com Velocidade de Aproximação

Este perfil (Figura 3.40) é o mais comum nos ascensores, pois garante o melhor acerto ao patamar (em relação aos apresentados anteriormente) e um conforto bastante aceitável para o utilizador. A Figura 3.41 é o logigramma utilizado para esta situação.

O código ST para este perfil está em: Apêndice A.

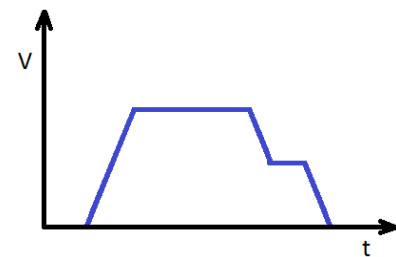


Figura 3.40: Perfil em rampa com velocidade de aproximação

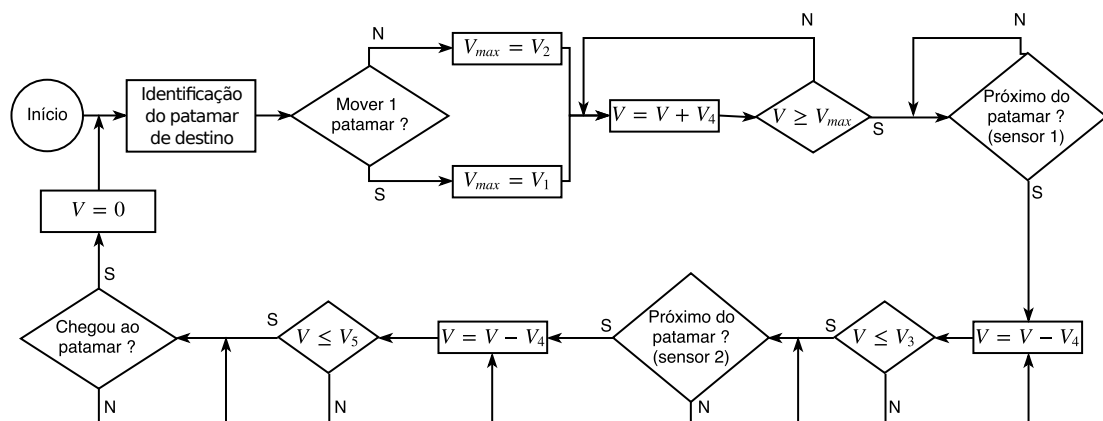


Figura 3.41: Logigramma para produzir o perfil em rampa com velocidade de aproximação

Perfil Definido em Função da Posição

Na Figura 3.42 o logigramma garante que a velocidade depende da posição da cabina. Neste caso, o perfil gerado acaba por se assemelhar ao perfil em rampa (da Figura 3.38).

No logigrama o bloco $V = V(P_{preferencia}, P_{atual})$ significa que a velocidade é definida em função da posição (P). Esta dependência é resolvida utilizando um controlador PI (Figura 3.43).

O código ST para este perfil está em: Apêndice B.

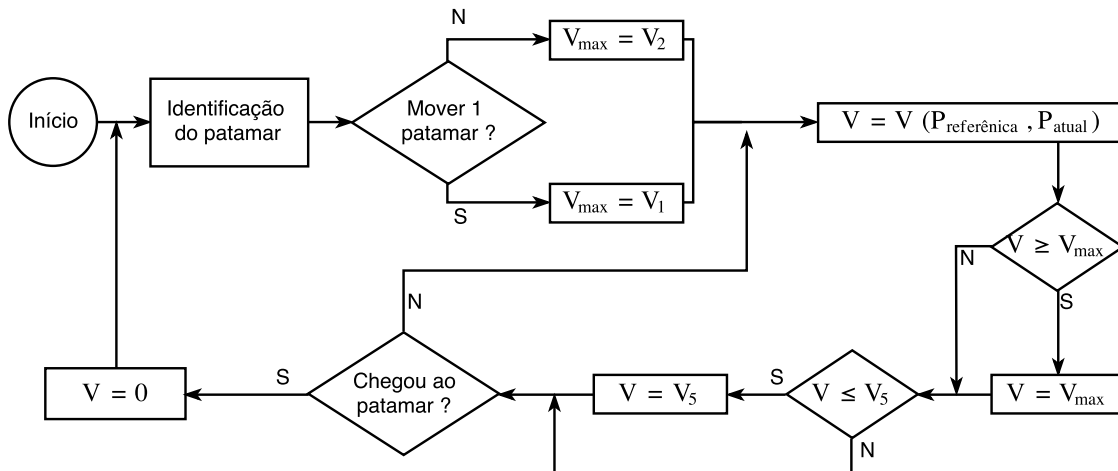


Figura 3.42: Logigrama para produzir o perfil definido em função da posição

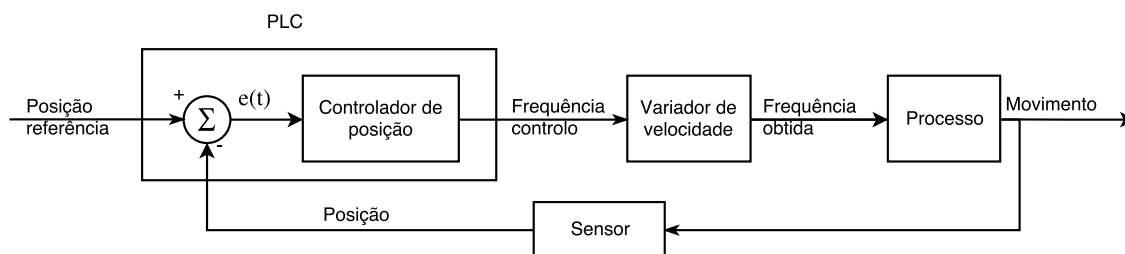


Figura 3.43: Anel de controle para o controle da velocidade em função da posição

Perfil com Limitação de Aceleração

O objetivo deste perfil é garantir o melhor conforto possível para o utilizador sem alterar a estrutura do elevador. Este perfil tem o objetivo de reduzir ao máximo as acelerações de modo a que se tornem impercetíveis para o utilizador. O logigrama utilizado está representado pela Figura 3.44. Esta dependência é resolvida utilizando um controlador PI (Figura 3.45).

O código ST para este perfil está em: Apêndice B.

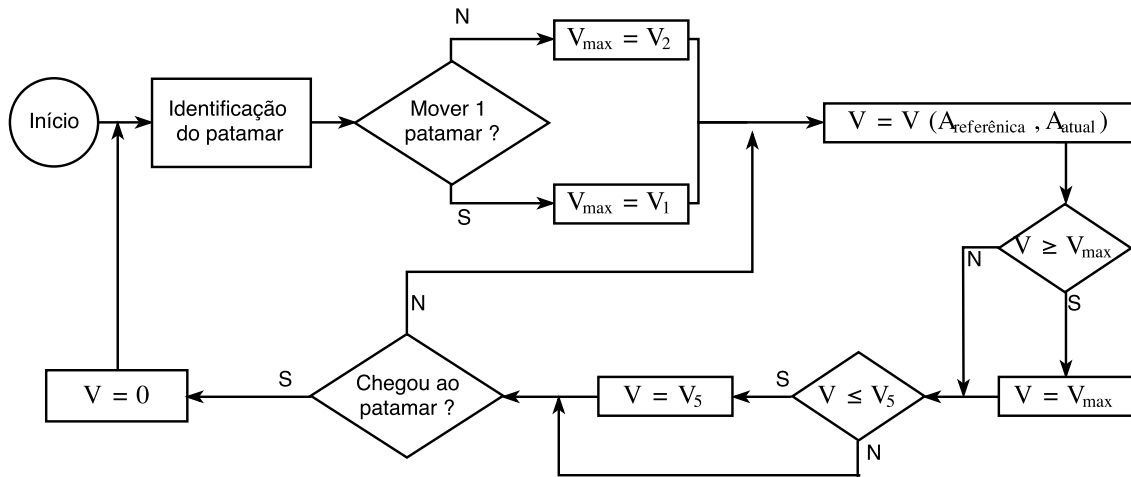


Figura 3.44: Logigrama para produzir o perfil com limitação de aceleração

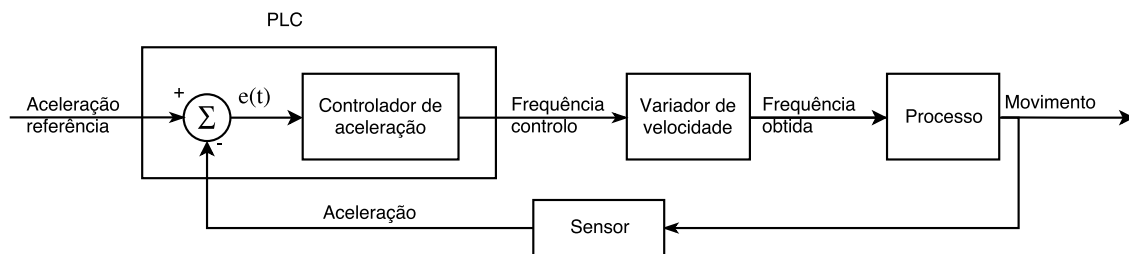


Figura 3.45: Anel de controlo para o controlo da aceleração

Capítulo 4

Resultados Experimentais

4.1 Introdução

O presente capítulo encontra-se dividido em seis partes. Na primeira são apresentados os resultados obtidos para o acelerómetro e para o barómetro. Na segunda os resultados obtidos pelo SM quando utilizado em ascensores públicos. A terceira parte é dedicada à instalação do sensor de ultrassons e do SM no elevador modelo. A quarta parte é relativa à modelação do elevador modelo. Na quinta parte são apresentados e avaliados os perfis de velocidade propostos. Na última parte são discutidos os resultados obtidos.

4.2 Resultados Obtidos para os Sensores

4.2.1 Acelerómetro

Após proceder à calibração do sensor, apresentada no capítulo 3, foi estudado de que modo a temperatura influenciava as medições (código no Apêndice D). Para tal, o acelerómetro foi colocado num ambiente frio até atingir a temperatura de 10°C, posteriormente foi colocado num ambiente a 26°C, onde permaneceu imóvel até atingir o equilíbrio térmico. Os resultados das acelerações foram registados em função da temperatura e são apresentado nas Figuras 4.1, 4.2 e 4.3.

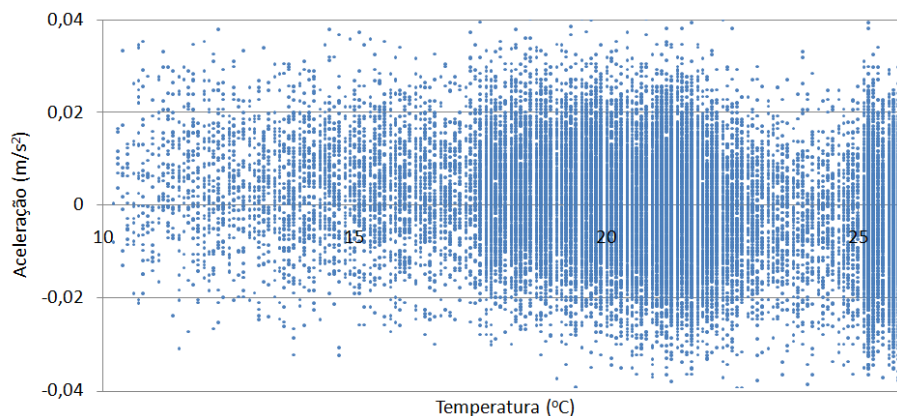


Figura 4.1: Variação de *bias* devido à temperatura no eixo X

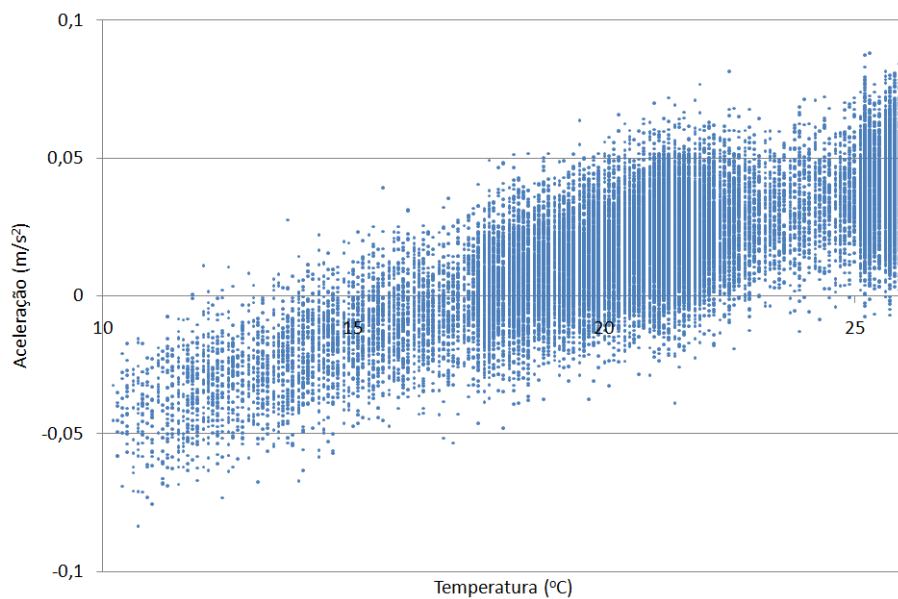


Figura 4.2: Variação de *bias* devido à temperatura no eixo Y

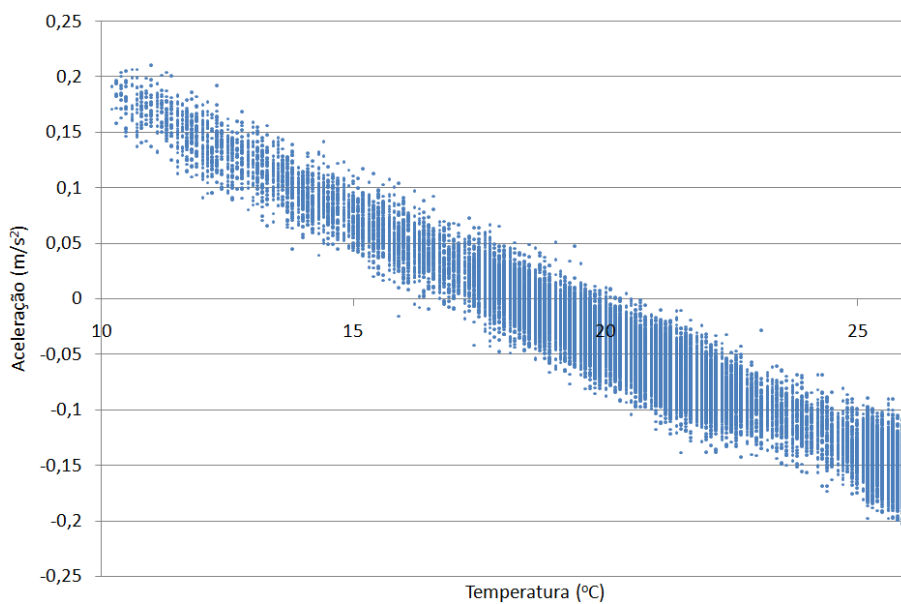


Figura 4.3: Variação de *bias* devido à temperatura no eixo Z

Depois de efetuar as correções associadas à temperatura, o MPU6050 foi colocado imóvel numa superfície horizontal durante cinco minutos e foi utilizada a técnica de Navegação Inercial (código no Apêndice D). Observou-se que os valores médios para os eixos x, y e z foram: -0.0003 , -0.0002 e -0.0004 m/s^2 .

Ao realizar a integração da aceleração para o eixo Z (Figura 4.4), eixo do movimento num ascensor, obtém-se a Figura 4.5. Para calcular a posição efetua-se uma nova integração e o resultado obtido é o apresentado na Figura 4.6. O acelerómetro manteve-se imóvel durante a recolha de dados, sendo o resultado esperado zero. Para contornar este problema foi considerado como aceleração nula o intervalo de -0.06 a 0.06 m/s^2 .

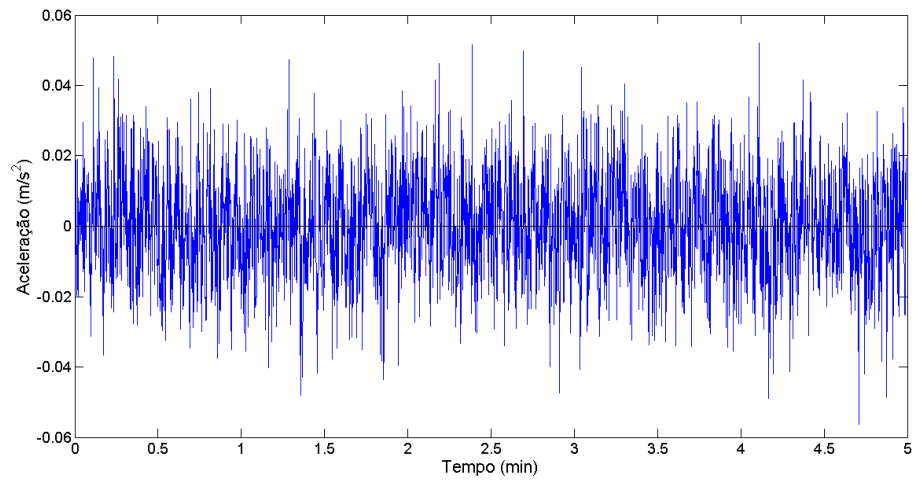


Figura 4.4: Resultado obtido no eixo Z após calibração

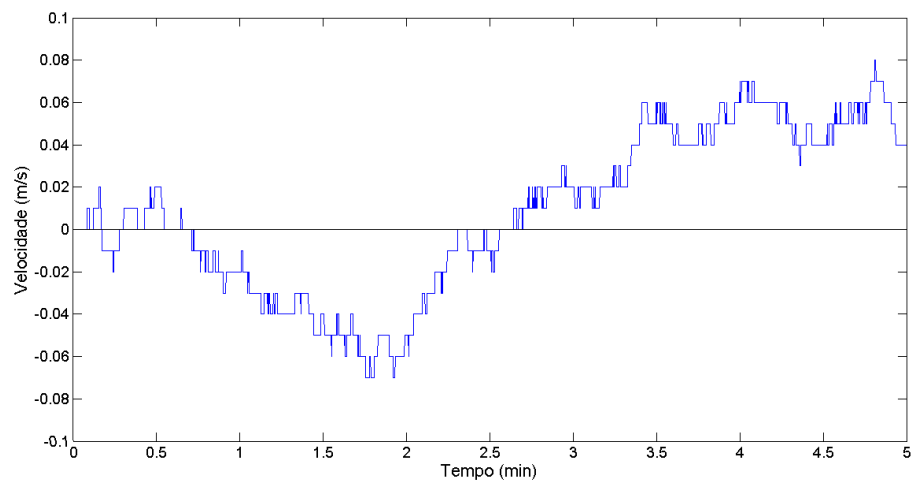


Figura 4.5: Integração dos resultados da Figura 4.4

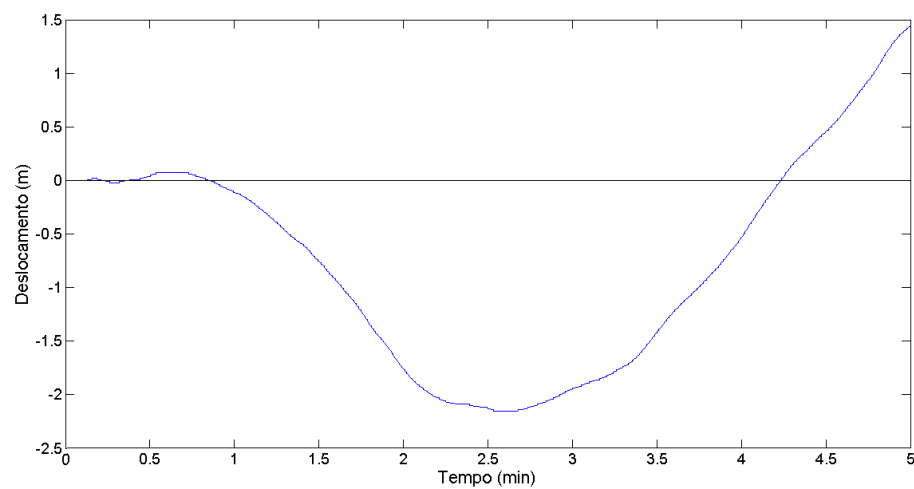


Figura 4.6: Dupla integração dos resultados da Figura 4.4

4.2.2 Barómetro

Utilizando as calibrações de fábrica para este sensor, e aplicando o filtro de Kalman aos resultados obtidos, é possível construir a Figura 4.7 (código no Apêndice D). O resultado a verde representa as medições efetuadas e a filtragem é representada a preto.

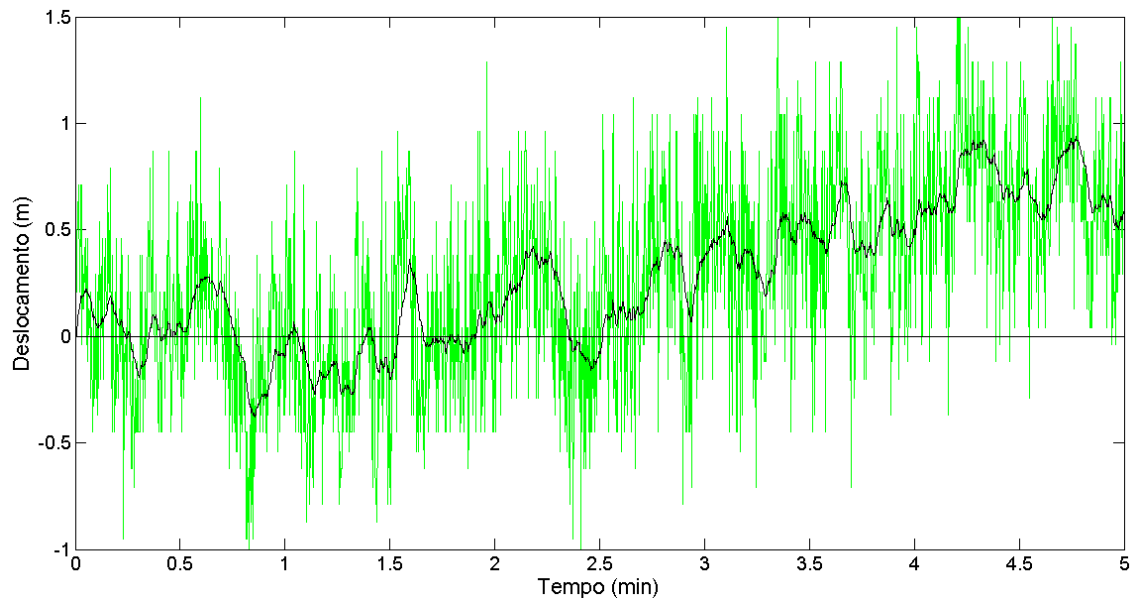


Figura 4.7: Barómetro imóvel com as calibrações de fábrica

4.3 Resultados Obtidos pelo Sistema de Medição em Ascensores Públicos

4.3.1 Determinação de Aceleração, Velocidade e Posição

O SM, colocado no elevador do DEE da FCT-UNL, efetuou trinta medições no seguinte percurso: patamar 2 - patamar 3 - patamar 2.

Foram estudadas duas situações. Na primeira (Figuras 4.8, 4.9 e 4.10) foi utilizado apenas o acelerómetro para efetuar as medições. Na segunda situação (Figuras 4.11 e 4.12) foram utilizados o acelerómetro e barómetro, código no Apêndice D.

Após as trinta medições foi efetuada a média. Esta é representada nas Figuras 4.8 a 4.12 pela linha “****”.

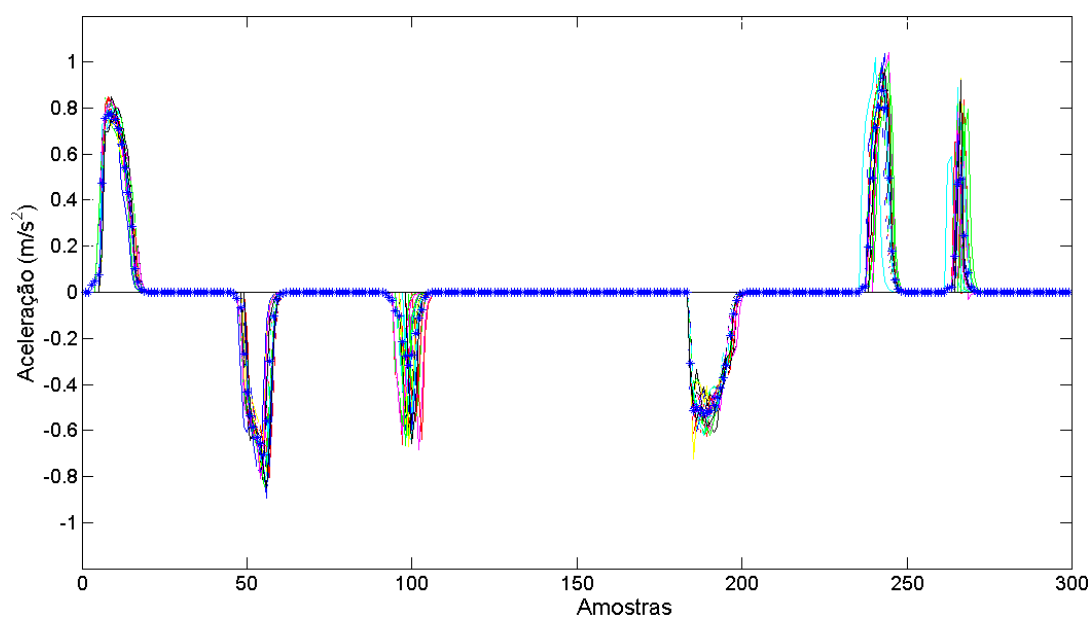


Figura 4.8: Trinta medições de aceleração utilizando acelerómetro

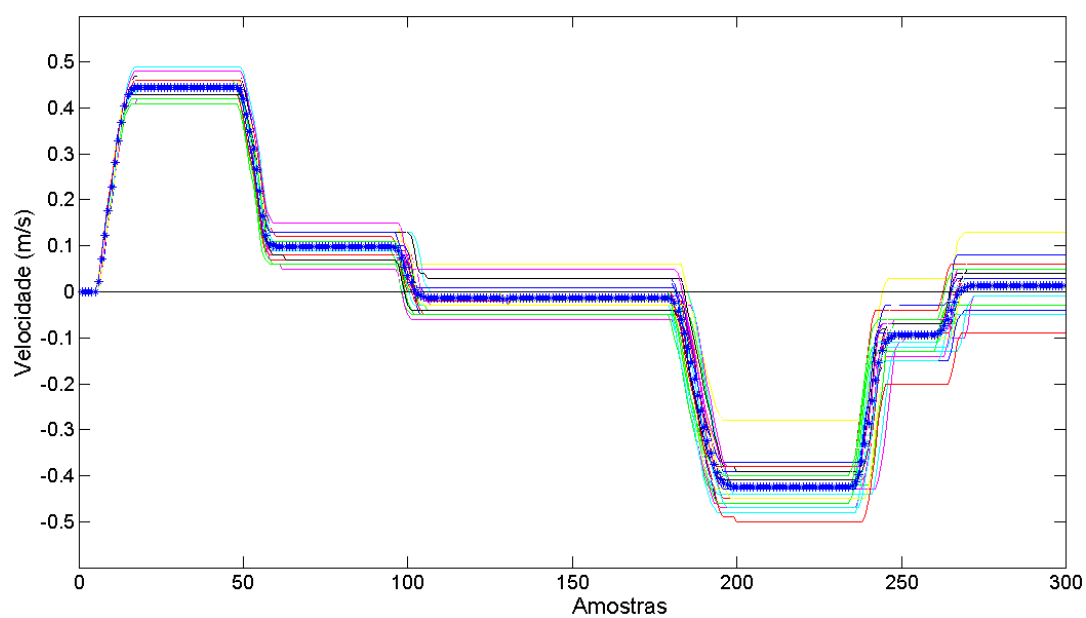


Figura 4.9: Trinta estimativas de velocidade utilizando acelerómetro

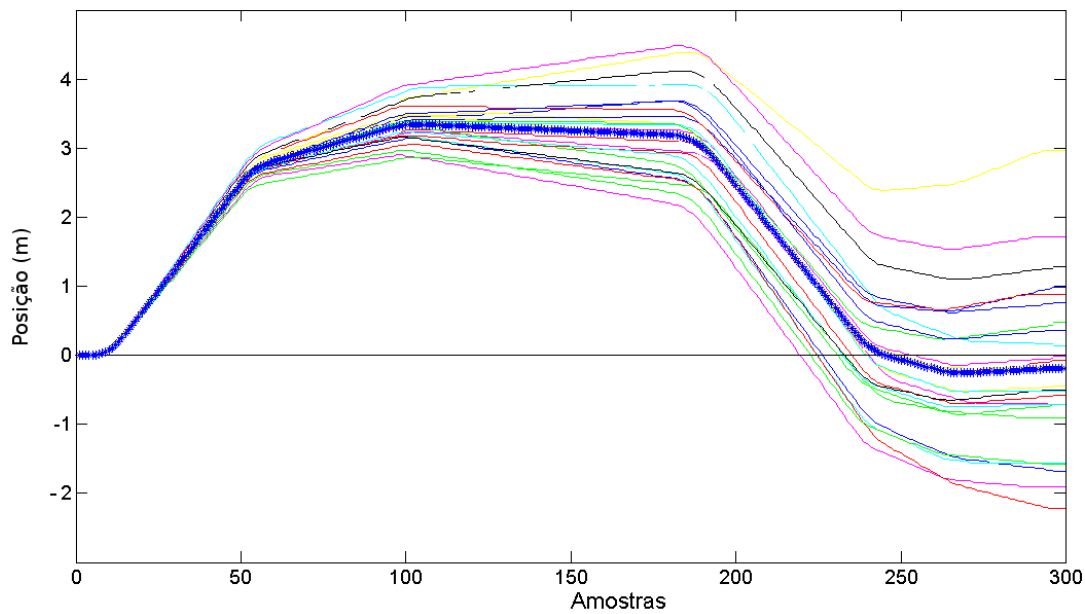


Figura 4.10: Trinta estimações de posição utilizando acelerômetro

Pela observação das figuras anteriores é possível identificar que, no início do movimento os resultados das trinta medições são muito próximos, mas com o decorrer da experiência a variância dos resultados aumenta. Este aumento ocorre devido à integração de pequenos erros que se propagam com o decorrer da experiência, tal como é evidenciado na Figura 4.10. Esta figura representa o deslocamento da cabina. No final da experiência, o resultado esperado seria zero o que não se verificou. Apesar de tudo, em média esta técnica consegue dar uma boa ideia dos perfis do elevador.

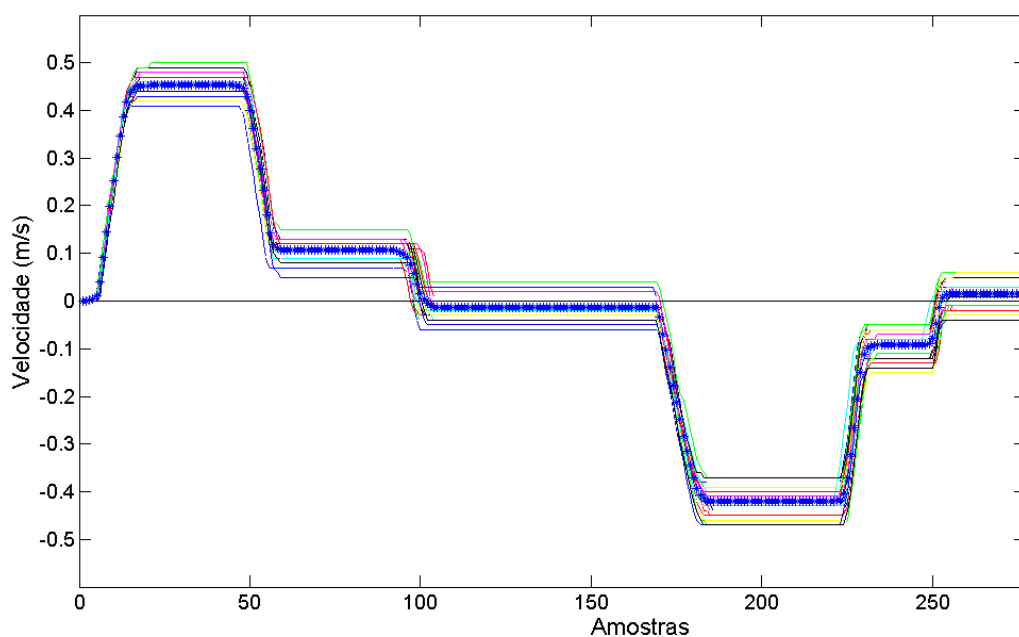


Figura 4.11: Trinta estimações de velocidade utilizando acelerômetro e barômetro

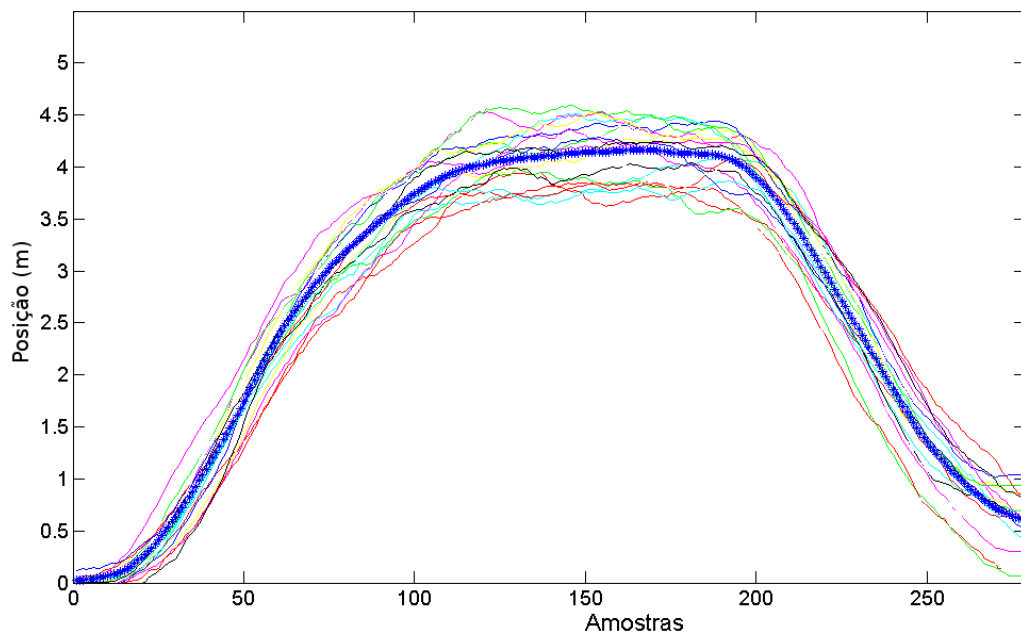


Figura 4.12: Trinta estimações de posição utilizando acelerómetro e barómetro

Com a introdução do barómetro os resultados da velocidade (Figura 4.11) melhoraram ligeiramente. Quanto à posição (Figura 4.12) registaram-se melhorias significativas. Note-se que após a introdução do barómetro a altura máxima rondava os 4 metros, enquanto no caso anterior a altura máxima era de 3.5 metros. Constatou-se com a aplicação de um simples teste de verificação que o valor real para a altura entre os patamares é de 3.9 m.

A utilização do barómetro compensou os erros relacionados com a integração da aceleração, reduziu a variância e forneceu estimações da posição muito melhores com um erro de apenas 500 mm.

4.3.2 Inquérito Sobre o Desempenho Ergonómico

O inquérito consistiu na participação de estudantes para fazerem duas viagens num ascensor: subida até ao último patamar, seguida de descida até ao primeiro. No fim de cada uma das viagens foi pedido aos estudantes que avaliassem a viagem de 1 a 5 para fazer comparação com o NDE (Tabela 3.3).

O estudo foi realizado em três ascensores de três edifícios do campus da FCT-UNL, a saber: civil, matemática e eletrotécnica. O estudo envolveu 30 estudantes universitários com idades compreendidas entre 20 e 26 anos. O estudo começou no edifício de civil o qual teve os piores resultados e acabou no edifício de eletrotécnica. O ascensor deste último edifício foi considerado o melhor dos três.

Em seguida são apresentados, para cada edifício, os resultados da aceleração vertical na subida e na descida, e a média e a variância dos resultados do inquérito. Os resultados r.m.s. e r.m.q. são os calculados pelo SM utilizando a tabela NDE (código no Apêndice D).

Optou-se por não apresentar os resultados das acelerações horizontais por não terem sido substancialmente relevantes, dado que nenhum dos inquiridos lhes fez referência.

Edifício de Civil

Dos três ascensores estudados este é o que impõe, durante o maior período de tempo, acelerações mais intensas (Figuras 4.13 e 4.15). Os estudantes referiram que sentiram um “salto” quando o elevador se aproximava do patamar de destino e quando parava. As Figuras 4.14 e 4.16 correspondem às avaliações dos estudantes.

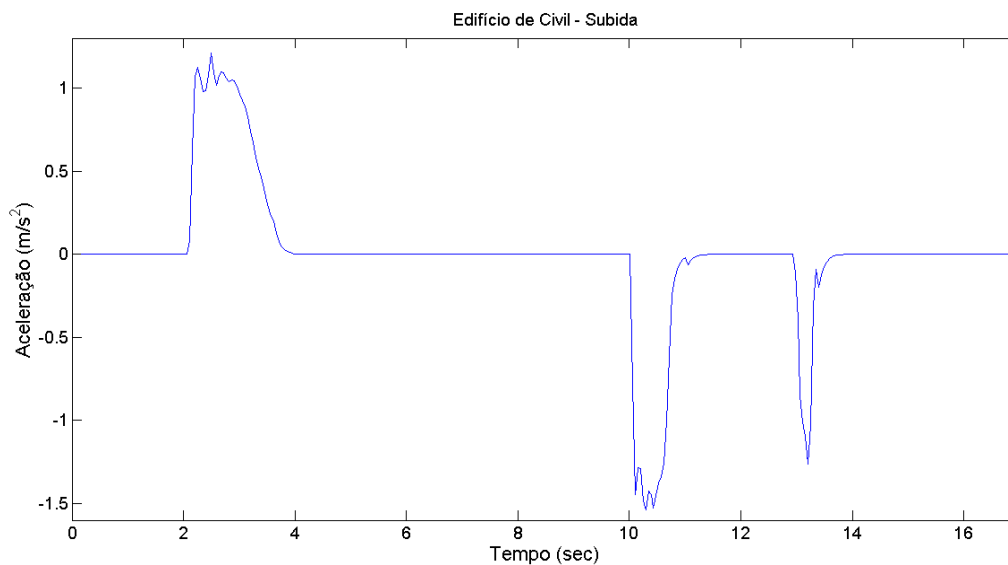


Figura 4.13: Perfil de aceleração do elevador do edifício de civil na subida



Figura 4.14: Resultados da avaliação do conforto do elevador do edifício de civil na subida

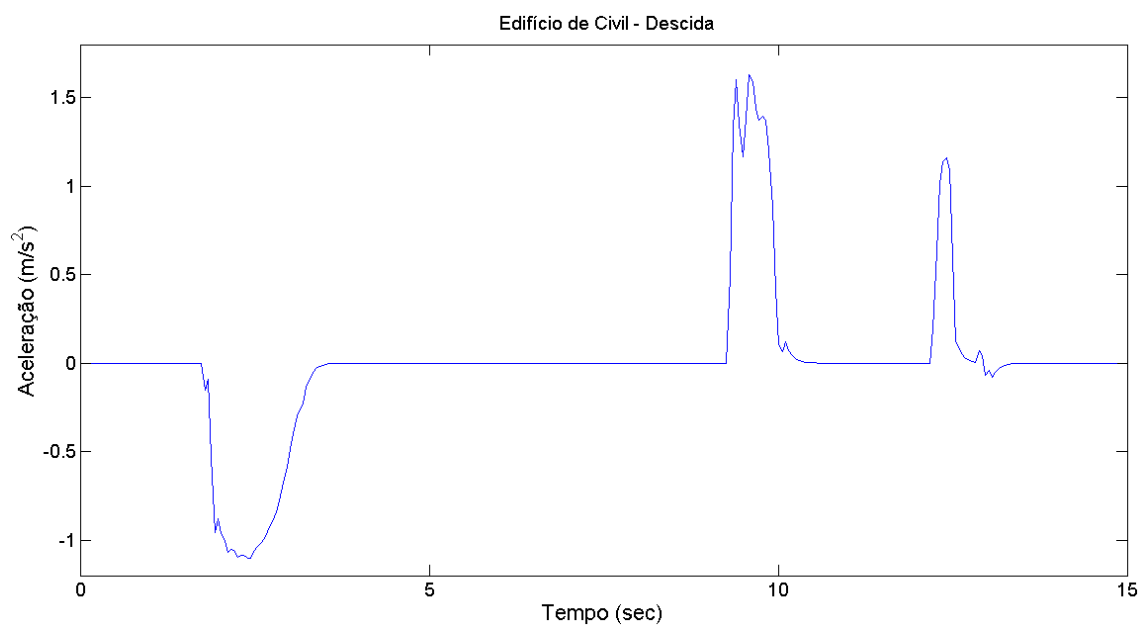


Figura 4.15: Perfil de aceleração do elevador do edifício de civil na descida



Figura 4.16: Resultados da avaliação do conforto do elevador do edifício de civil na descida

Edifício de Matemática

O perfil de aceleração deste ascensor é muito diferente dos apresentados até agora. O início é suave (especialmente na descida Figura 4.19) mas a paragem é extremamente repentina (Figuras 4.17 e 4.19). Com efeito, os estudantes não sentiram nenhum “salto” como no ascensor de civil, o que se refletiu na avaliação deste ascensor (Figuras 4.18 e 4.20).

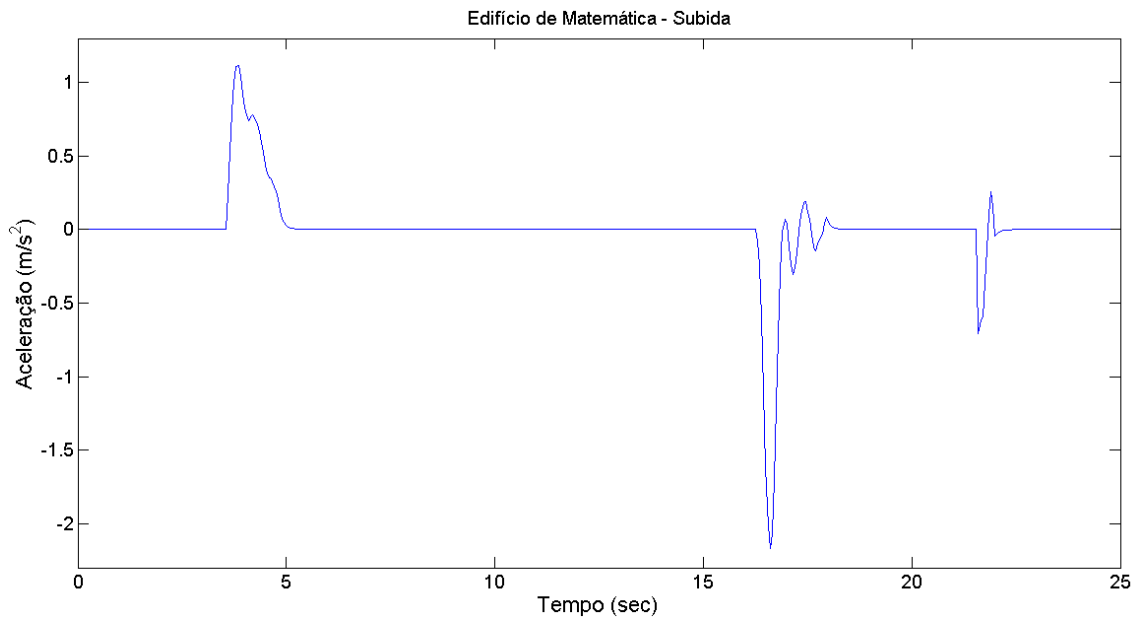


Figura 4.17: Perfil de aceleração do ascensor do edifício de matemática na subida

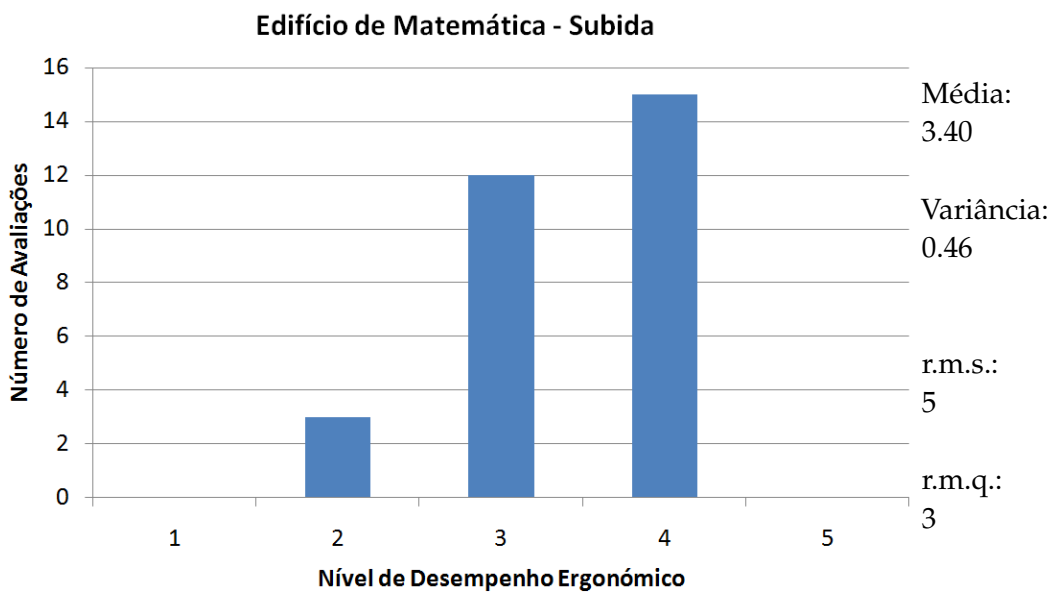


Figura 4.18: Resultados da avaliação do conforto do ascensor do edifício de matemática na subida

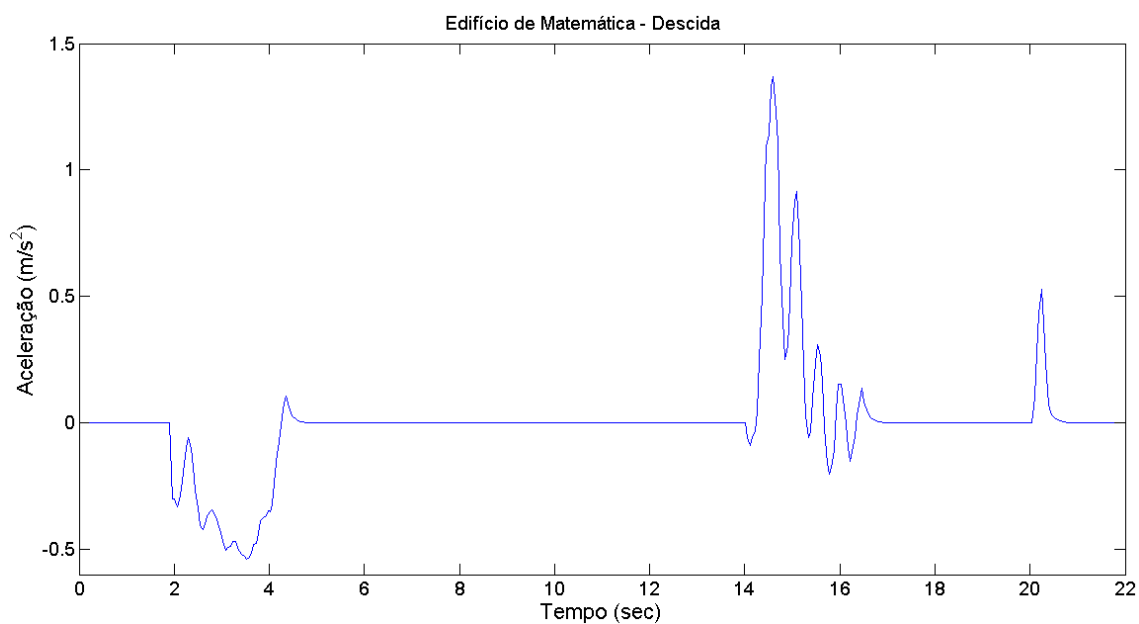


Figura 4.19: Perfil de aceleração do elevador do edifício de matemática na descida

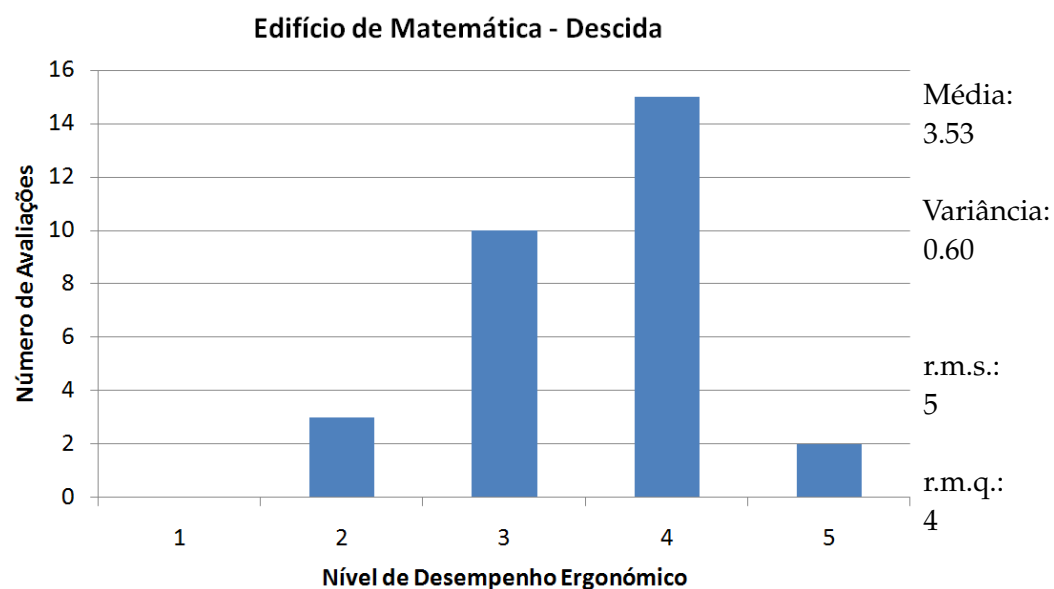


Figura 4.20: Resultados da avaliação do conforto do elevador do edifício de matemática na descida

Edifício de Eletrotécnica

Este foi o último ascensor visitado pelos estudantes. Como se pode observar nas Figuras 4.21 e 4.23 as intensidades são muito inferiores às dos restantes ascensores. Como consequência registou as melhores classificações (Figuras 4.22 e 4.24).

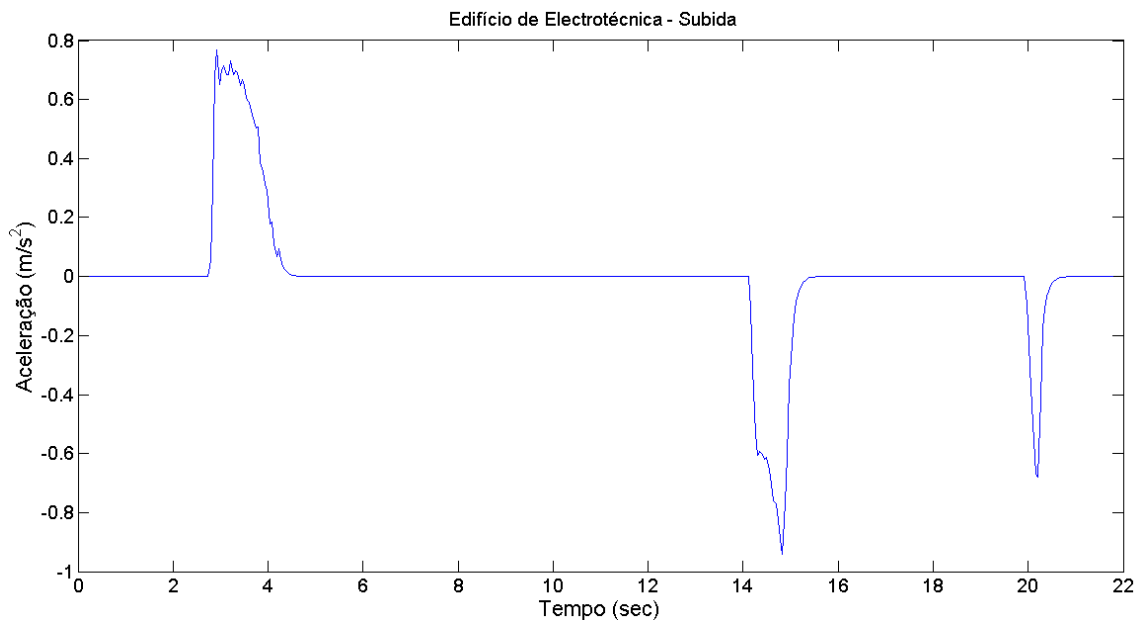


Figura 4.21: Perfil de aceleração do ascensor do edifício de eletrotécnica na subida

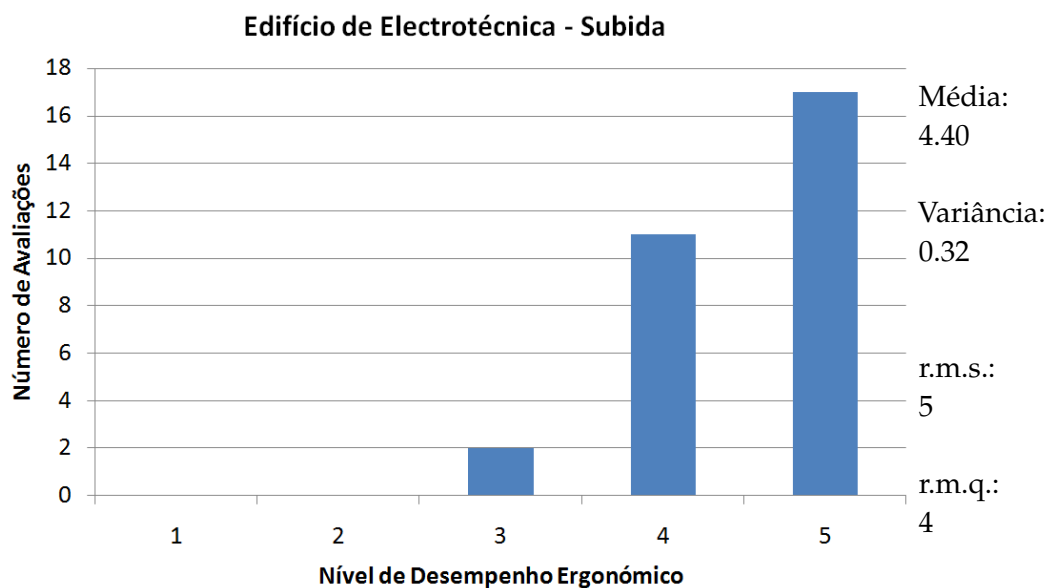


Figura 4.22: Resultados da avaliação do conforto do ascensor do edifício de eletrotécnica na subida

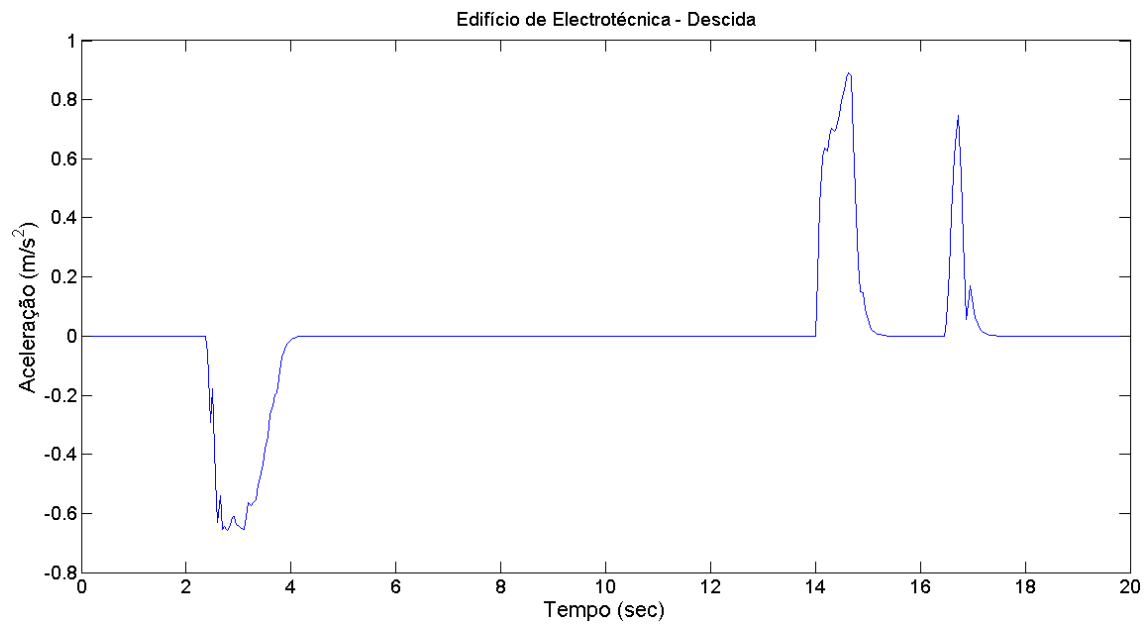


Figura 4.23: Perfil de aceleração do elevador do edifício de eletrotécnica na descida

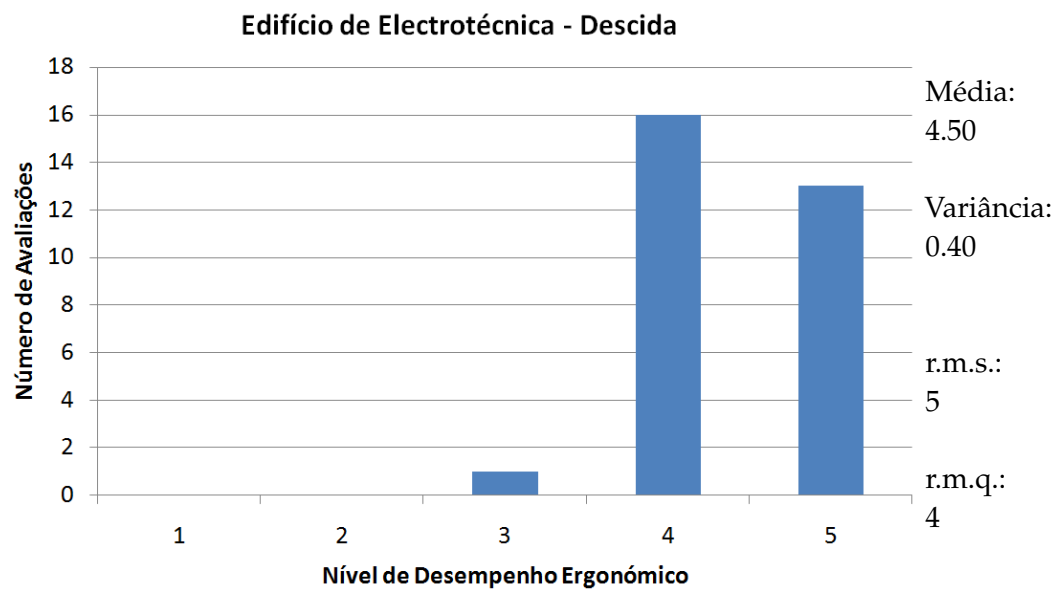


Figura 4.24: Resultados da avaliação do conforto do elevador do edifício de eletrotécnica na descida

Verifica-se que, ao utilizar o método r.m.q e compará-lo com o NDE obtém-se exatamente o resultado médio arredondado à unidade dos inquiridos (para cinco dos seis inquiridos).

4.4 Teste dos Sensores no Modelo de Elevador

Saber a posição exata da cabina e a aceleração a que é sujeita são fatores importantes quando se aborda o tema de viagem ergonómica. O elevador modelo não fornece informação sobre estes fatores, sendo necessário instalar sensores adicionais.

4.4.1 Sensor de Distância Ultrassónico

Instalação

O sensor foi instalado na caixa do elevador de modo a medir a posição do contrapeso (código no Apêndice D). Optou-se por esta solução porque os veios roscados utilizados para fixar o motor na estrutura estavam disponíveis (Figura 4.25(a)).

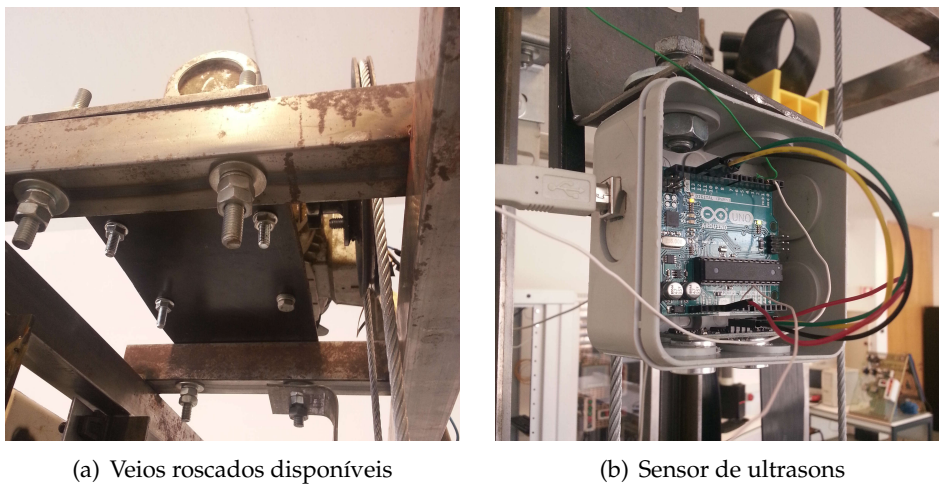


Figura 4.25: Pormenores relativos à instalação do sensor de ultrassons

Este sensor é ligado a uma placa Arduino UNO. A informação é enviada por PWM para um filtro passa baixo de segunda ordem e posteriormente segue para o PLC onde é recebida no módulo AMZ600, entrada 2, endereço %IW5.1. O hardware encontra-se dentro de uma caixa (Figura 4.25(b)). O Arduino é alimentado via porta USB por um computador, permitindo ainda a comunicação série entre o sensor e o computador.

Para refletir o som enviado pelo sensor foi introduzida uma placa de acrílico com 4 mm de espessura no contrapeso (Figura 4.26). A estrutura, onde foi colocado o sensor, permite regular a distância entre este e o contrapeso.

Teste

Os dados das Figura 4.27(a) e 4.27(b) foram recolhidos por uma placa de aquisição de dados (NI USB-6008) ligada na entrada 8 da placa AMZ600. As figuras representam as medições efetuadas pelo sensor de ultrassons consoante o perfil de velocidade da cabina. As linhas horizontais verdes representam os patamares. Como se pode observar nas figuras, o sensor é sensível a diferentes métodos de aproximação ao patamar.



Figura 4.26: Pormenor do sensor de ultrassons e placa de acrílico no contrapeso

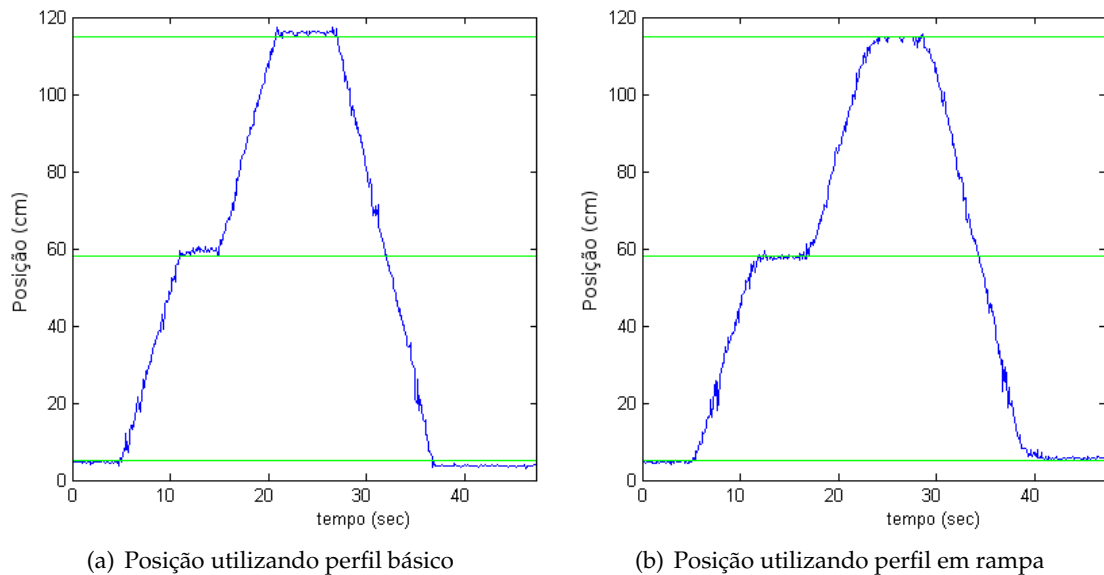


Figura 4.27: Medição em centímetros do deslocamento da cabine em função do tempo em segundos

Interpretação dos Valores Enviados (na ótica do PLC)

A Equação 4.1 converte tensão para centímetros.

$$\text{Centímetros} = 0.0339752 * \text{Volt} - 9.689 \quad (4.1)$$

Conclui-se que o sensor de distância ultrassónico têm sensibilidade suficiente para distinguir diferentes formas de acerto ao patamar.

4.4.2 Sistema de Medição

Instalação

O SM é colocado no interior da cabina como mostra a Figura 4.28. O sensor é ligado ao PLC pela entrada 6 da placa AMZ600, que corresponde ao endereço %IW5.3. Nesta, lê-se a aceleração vertical sentida na cabina. Apenas é enviada a aceleração porque o elevador tem mecanismos que efetuam melhores medições da velocidade e da posição da cabina.



Figura 4.28: Instalação do SM na cabina do elevador modelo

Teste

Com o intuito de conhecer o comportamento do SM, quando utilizado no elevador modelo, foram registados e comparados os perfis de posição e velocidade para um percurso específico da cabina. O percurso consistiu em, partindo do patamar 0 e utilizando o perfil básico, deslocar a cabina para o patamar 1, onde permaneceu no patamar durante cinco segundos. Subiu em seguida para o patamar 2, no qual aguardou outros cinco segundos antes de voltar para o patamar 0.

Na Figura 4.29 regista-se a comparação do deslocamento utilizando os ultrassons (em cima) e utilizando o SM (em baixo). Na Figura 4.30 é feita a comparação entre a velocidade estimada pelo SM (em baixo) e a tensão à saída do variador de velocidade (em cima). A conversão de tensão para velocidade é efetuada recorrendo à Equação (3.12).

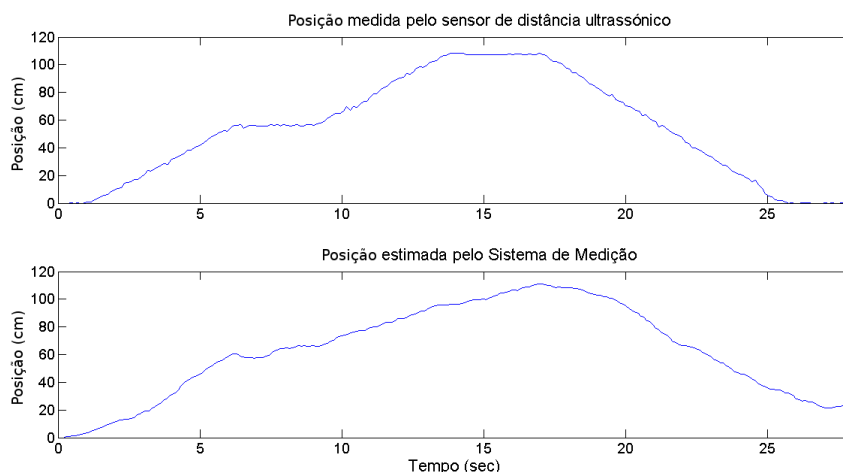


Figura 4.29: Medição do deslocamento utilizando ultrassons e o SM

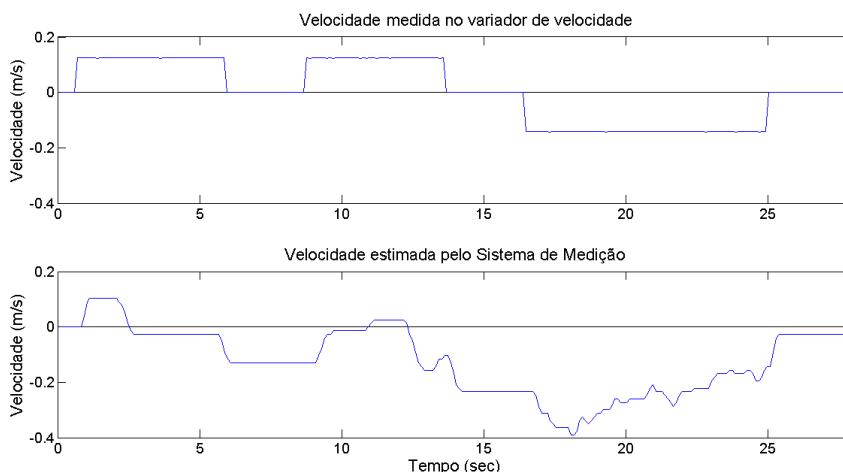


Figura 4.30: Medição da velocidade utilizando o variador de velocidade e o SM

Como se pode observar nas figuras anteriores, a utilização dos sensores existentes no elevador tem uma melhor desempenho do que o SM. Deste modo, o SM apenas envia a aceleração. Como se pode observar pela Figura 4.31 no envio da informação é introduzido ruído.

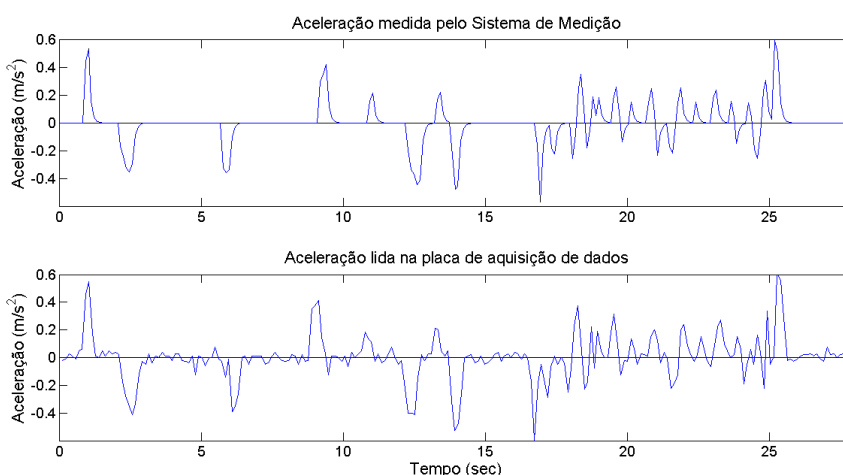


Figura 4.31: Comparação entre a medição e receção da aceleração

Interpretação dos Valores Enviados (na ótica do PLC)

A Equação 4.2 converte tensão para metros por segundo quadrado.

$$\text{Aceleração [m/s}^2\text{]} = 0.00941 * \text{Tensão [V]} - 1.93 \quad (4.2)$$

4.5 Perfis Gerados

Nas Figuras 4.32 a 4.37 são apresentados os resultados para cada um dos seis perfis sugeridos. Na Figura 4.38 pretende-se representar o salto de um passageiro durante a viagem e a respetiva resposta do elevador, quando utilizado o sexto perfil.

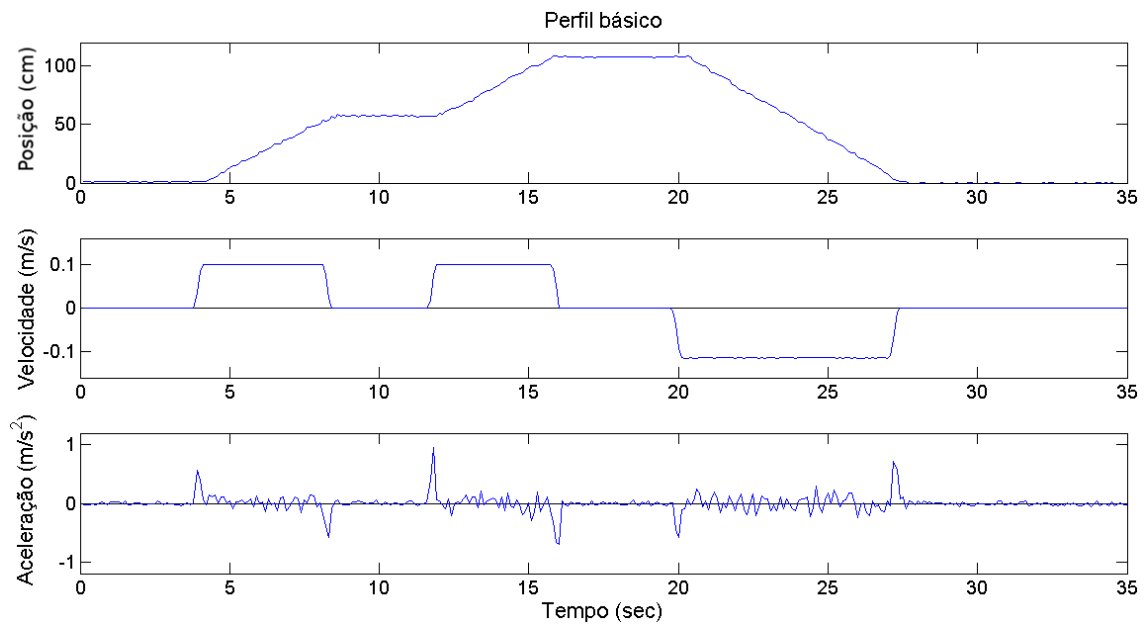


Figura 4.32: Perfil 1, básico

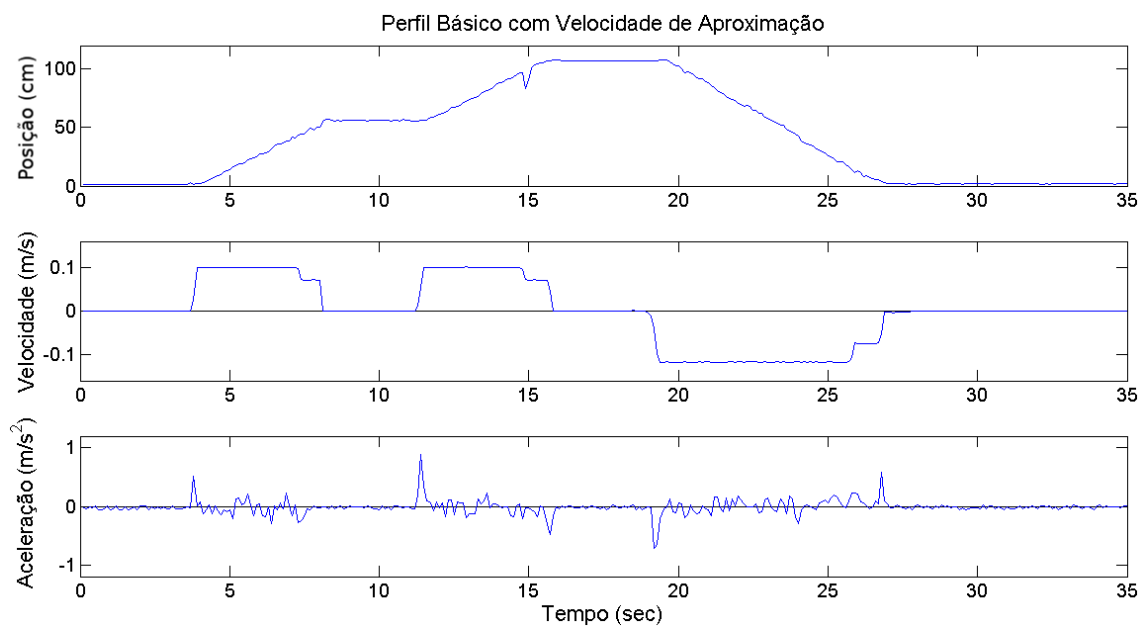


Figura 4.33: Perfil 2, básico com velocidade de aproximação

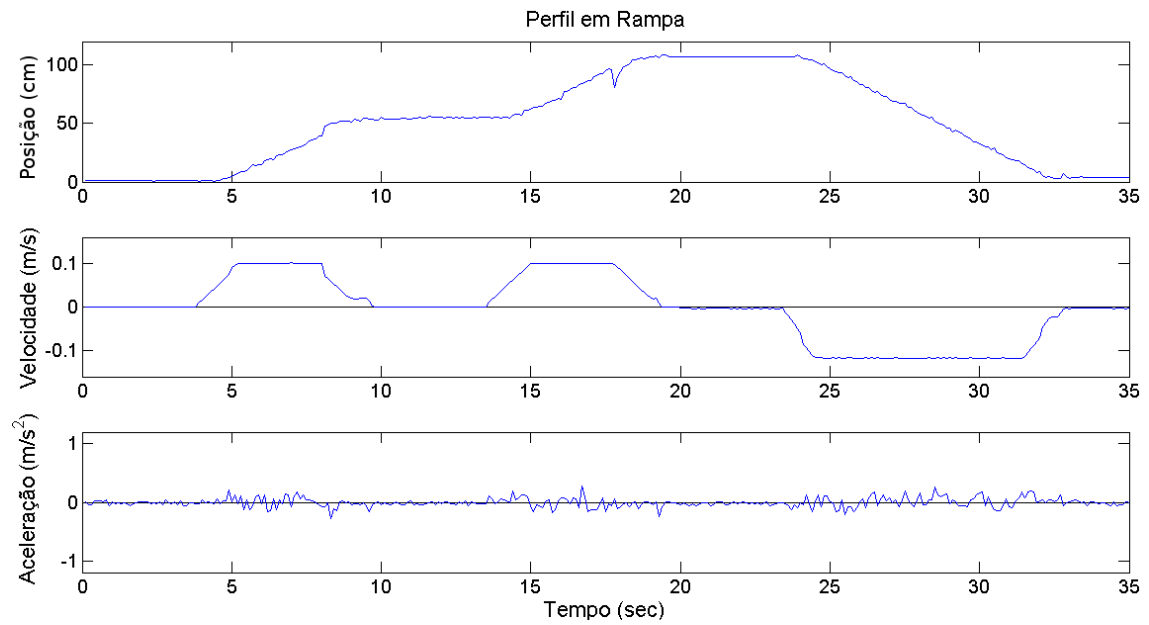


Figura 4.34: Perfil 3, rampa

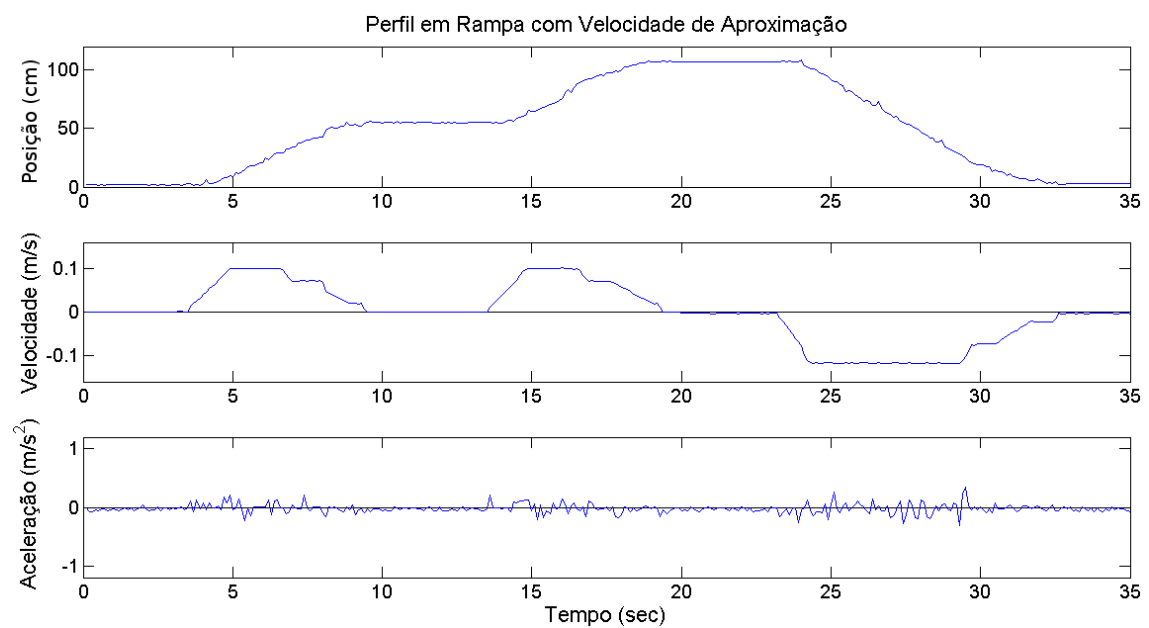


Figura 4.35: Perfil 4, rampa com velocidade de aproximação

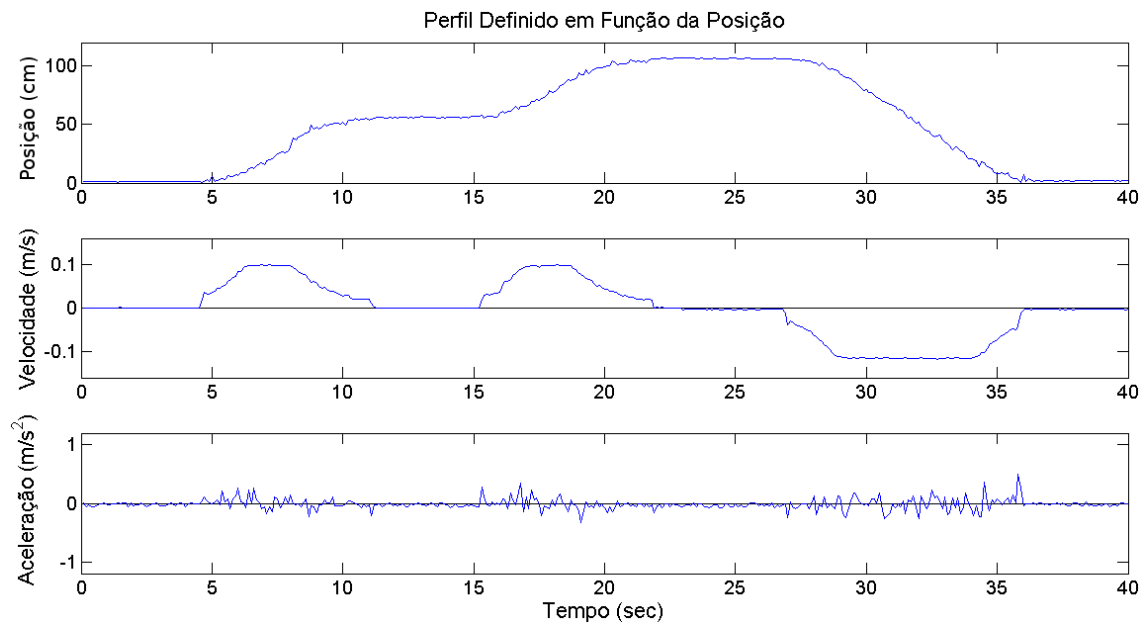


Figura 4.36: Perfil 5, definido em função da posição

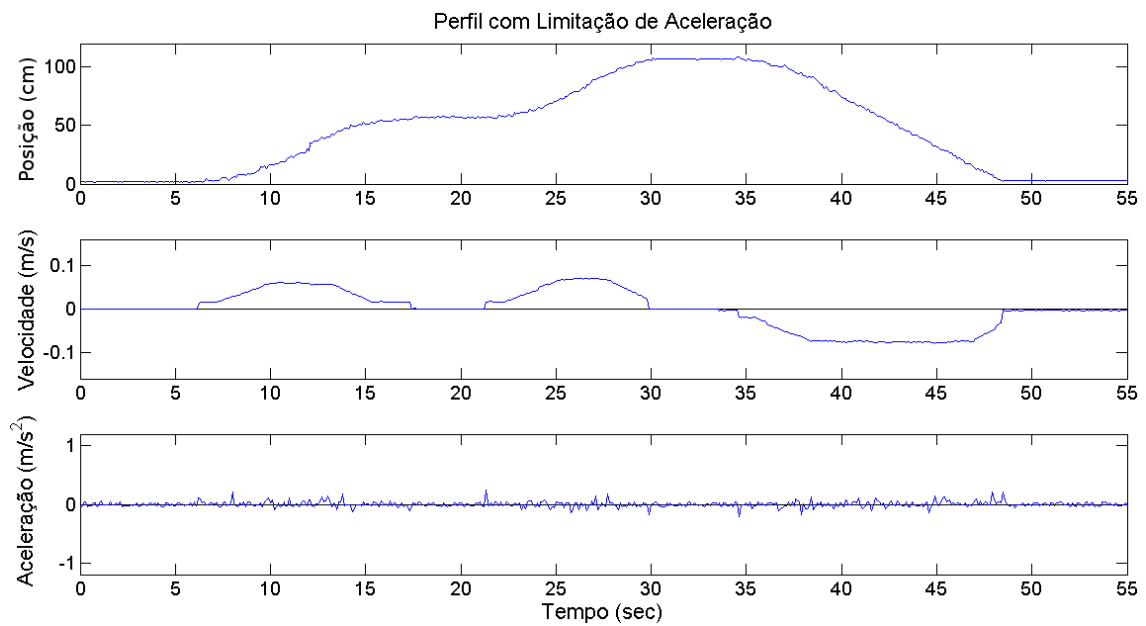


Figura 4.37: Perfil 6, com limitação de aceleração

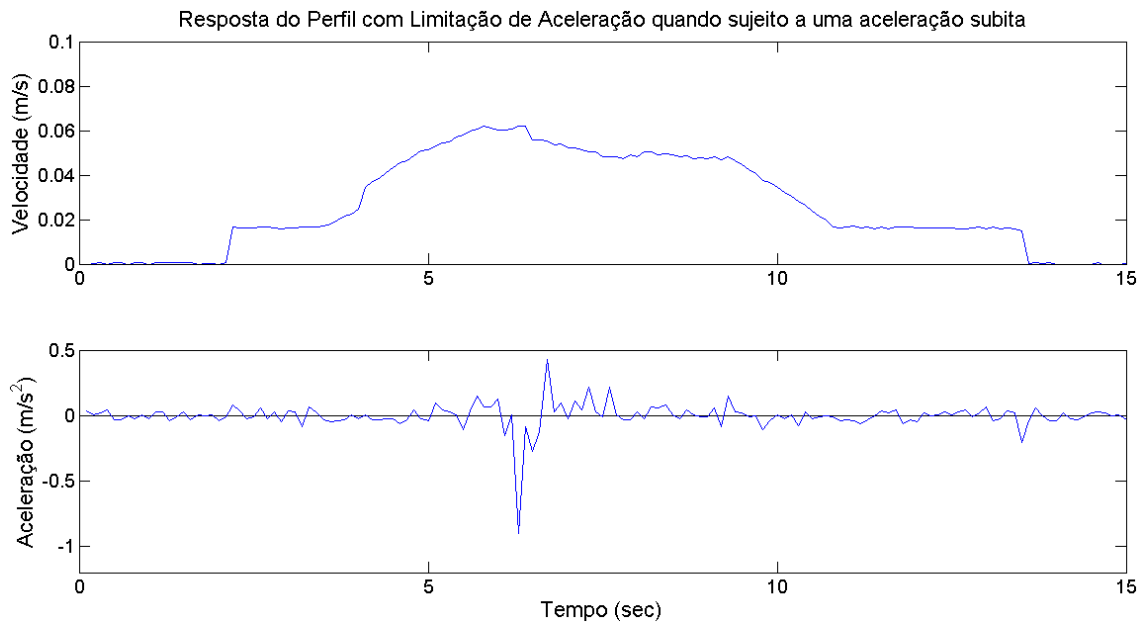


Figura 4.38: Simulação de um salto na cabina utilizando o perfil 6

As acelerações sentidas na cabina são muito inferiores às experienciadas nos elevadores do nosso quotidiano, deste modo os limites utilizados no NDE não têm significado.

Para efetuar a avaliação ergonómica dos perfis foi utilizado o método r.m.q.. Consoante os resultados obtidos os perfis foram ordenados do melhor para o pior (Tabela 4.1). Verificou-se que o perfil 6 (perfil com limitação de aceleração) é o melhor.

	Melhor ... Perfil ... Pior					
Subida	6	4	3	5	2	1
Descida	6	4	3	5	2	1

Tabela 4.1: Avaliação ergonómica dos seis perfis propostos

4.6 Discussão de Resultados

Utilizar apenas o acelerómetro para determinar o deslocamento na cabina não é suficiente porque, como se observa na Figura 4.12, as acelerações medidas não foram exatamente iguais. Estas pequenas discrepâncias a longo prazo levam a que o resultado final tenha uma grande variância (Figura 4.10). A introdução de um barómetro permite obter um melhor resultado para a posição da cabina (Figura 4.12). Em média, o SM efetua uma boa estimativa dos perfis da cabina de um ascensor.

Foi escolhido iniciar o inquérito no ascensor do edifício de civil por ter valores elevados

de aceleração. A maior parte dos estudantes avaliou-o com a nota 3, mas depois de utilizar o ascensor de matemática, alguns, reduziram a nota para 2. Quanto mais informação for dada ao inquirido, melhor é o referencial de conforto que ele cria e mais concisa é a sua resposta no inquérito.

As estimações dos perfis, efetuadas pelo SM no elevador modelo, não são boas. A máquina de tração foi fixa na mesma estrutura onde a cabina se desloca e, devido à falta de amortecimento, as vibrações provocadas pela máquina de tração são transmitidas para a cabina. Deste modo o SM não consegue diferenciar aceleração de vibração, não produzindo boas estimações no elevador modelo. Para o SM diferenciá-las seria necessário conhecer a sua frequência. Para determinar a frequência seria necessário recorrer a um osciloscópio, um vez que o Arduino não consegue efetuar transformadas rápidas de Fourier. Para utilizar o SM no elevador modelo, ou se instala um osciloscópio ou se amortiza as vibrações provocadas pela máquina de tração.

A frequência de aquisição dos dados (utilizados para a construção do modelo), aliada as características não linear do elevador, resultou em más simulações dos controladores (Figuras 3.30 e 3.31).

O perfil com limitação de aceleração demora mais tempo a desloca-se entre patamares. Este aumento de duração na viagem surge porque não é possível diferenciar vibração de aceleração. Deste modo a velocidade máxima atingida é a que produz a menor intensidade de vibrações na cabina. Como observado na Figura 4.38, este perfil reage a choques ocasionais como por exemplo: um passageiro a saltar dentro da cabina.

Capítulo 5

Conclusões e Trabalho Futuro

Este capítulo identifica as conclusões obtidas no desenvolvimento desta dissertação e apresenta propostas de trabalhos futuros para a utilização do Sistema de Medição e para o elevador protótipo.

5.1 Conclusões

- Todos os objetivos propostos foram cumpridos;
- Mostrou-se ser possível melhorar e avaliar o desempenho ergonómico em ascensores.
- Para utilizar a técnica de navegação inercial, com base nos sensores de baixo custo utilizados, calibrar o acelerómetro e fazer o zero dos sensores são procedimentos fundamentais para obter bons resultados;
- Os resultados obtidos por navegação inercial devem ser corrigidos periodicamente, caso contrário estes deixam de ser relevantes;
- De acordo com as respostas ao inquérito, o NDE é válido para a avaliação do conforto em ascensores;
- A modelação do elevador não se verificou eficaz para projetar controladores;
- As características do elevador protótipo, aliadas à qualidade dos sensores utilizados, não permitam que o Sistema de Medição faça estimativas viáveis da velocidade no elevador protótipo.

5.2 Trabalho futuro

- Utilizar o resultado dos ultrassons e do acelerómetro para estimar a velocidade;
- Utilizar outro tipo de acelerómetro, por exemplo o ADXL, e comparar o desempenho com o MPU6050;

- Utilizar acelerómetro, giroscópio, introduzir bússola e GPS para transformar o Sistema de Medição num Sistema de Navegação. O Sistema de Navegação pode ser utilizado em diferentes meios de transporte, por exemplo: automóvel ou comboio;
- Estudar o elevador protótipo em termos de vibrações e propor métodos para reduzi-las na cabina;
- Instalar no chão da cabina um sensor de pressão/balança e avaliar o conforto em função do aumento da carga;
- Elaborar um estudo sobre o índice de desempenho do Sistema de Medição quando se calcula os perfis;
- Utilizar um sistema para calcular, electronicamente, as frequências de vibração e aplicar os filtros de ponderação em frequência às vibrações medidas.
- Otimização do elevador protótipo sujeito a dois constrangimentos: maximização do conforto na cabina e minimização da corrente consumida pela máquina de tração;

Bibliografia

- [1] *Design of S-curve Direct Landing Position Control System for Elevator Using Microcontroller*. Vol. 2. 2012.
- [2] M. J. Griffin. *Handbook of Human Vibration*. Elsevier Academic Press, 1990.
- [3] S. S. Rao. *Mechanical Vibrations*. Prentice Hall, 2011.
- [4] ISO 2631-1. *Mechanical vibration and shock - Evaluation of human exposure to whole-body vibration - Part 1 : General requirements*. 1997.
- [5] D. dos Remédios Margarido. “Metodologias para a Avaliação dos Níveis de Vibração em Estruturas do Ponto de Vista do Conforto Humano”. Tese de mestrado. Faculdade de Engenharia da Universidade do Porto, 2013.
- [6] B. S. Portela. “Vibração de corpo inteiro em motoristas de ônibus: associação com variáveis de aptidão física e dor lombar”. Tese de mestrado. Universidade Federal do Paraná, 2014.
- [7] DL 46/2006. *Prescrições mínimas de segurança e saúde respeitantes à exposição dos trabalhadores aos riscos devidos a vibrações mecânicas*. 2006.
- [8] S. A. Clevenson, T. K. Dempsey e J. D. Leatherwood. *Effect of Vibration Duration on Human Discomfort*. Rel. téc. 1283. NASA, 1978.
- [9] Y. Marjanen. “Validation and improvement of the ISO 2631-1 (1997) standard method for evaluating discomfort from whole-body vibration in a multi-axis environment”. Tese de doutoramento. Loughborough University, 2010.
- [10] ISO 8041. *Human response to vibration - Measuring instrumentation*. 2005.
- [11] ISO 18738(E). *Lifts (elevators) — Measurement of lift ride quality*. 2003.
- [12] G. C. Barney e L. Al-Sharif. *Elevator traffic handbook: theory and practice*. Routledge, 2003.
- [13] R. Howkins. “Elevator ride quality—the human ride experience”. Em: *Proceedings of the ELEVCON 2006, Elevator Technology* 16 (2006).
- [14] T. Boulter. “Elevator modeling and DC drive speed controller design”. Em: *Applied Industrial Control Solutions LLC* (2000).
- [15] *A General Survey on Lift Ride Quality at Public Buildings of the Hong Kong Special Administrative Region*. 2004.

- [16] Elevadores Atlas Schindler. *Manual de Transporte Vertical em Edifícios Elevadores de Passageiros, Escadas Rolantes, Obra Civil e Cálculo de Tráfego*. 2008.
- [17] T. Frey. "Enhanced elevator drive software improves ride quality, eases setup". Em: *ELEVATOR WORLD* 48.1 (2000), pp. 58–84.
- [18] B. Drury. *Control techniques drives and controls handbook*. The Institution of Engineering e Technology, 2009.
- [19] JL Seagull Power Products. *Elevator products details*. <http://www.pmtvib.com>. Verificação final: Março 2016.
- [20] AccuDIY.com. *HC-SR04 Ultrasonic Range Finder - Manual*. http://www.accudiy.com/download/HC-SR04_Manual.pdf. Verificação final: Março 2016.
- [21] Cytron Technologies Sdn. Bhd. *Product User's Manual - HC-SR04 Ultrasonic Sensor*. https://docs.google.com/document/d/1Y-yZnNhMYy7rwhAgyL_pfa39RsB-x2qR4vP8saG73rE/edit. Verificação final: Março 2016.
- [22] M. Cavcar. "The International Standard Atmosphere (ISA)". Em: *Anadolu University* 26470 (2000).
- [23] Bosch Sensortec. *Data sheet BMP180*. <https://www.adafruit.com/datasheets/BST-BMP180-DS000-09.pdf>. Verificação final: Março 2016.
- [24] InvenSense Inc. *MPU-6000 and MPU-6050 Product Specification Revision 3.3*. <http://www.seeedstudio.com/wiki/images/b/b1/MPU6050.pdf>. Verificação final: Março 2016.
- [25] O. J. Woodman. "An introduction to inertial navigation". Em: *University of Cambridge, Computer Laboratory, Tech. Rep. UCAMCL-TR-696* (2007).
- [26] J. G. Webster. *The Measurement, Instrumentation, and Sensors Handbook*. CRC press LLC, 1999.
- [27] N. Aboelmagd, T. B. Karmat e J. Georgy. *Fundamentals of Inertial Navigation, Satellite-based Positioning and their Integration*. Springer: Heidelberg, Germany, 2013.
- [28] C. C. Tsang. "Error reduction techniques for a MEMS accelerometer-based digital input device". Tese de doutoramento. The Chinese University of Hong Kong, 2008.
- [29] S. H. Bogdan Zak. "The Use of MEMS Accelerometers for Control of a Small Unmanned Underwater Vehicle". Em: (2014).
- [30] P. Gil. *Filtro de Kalman Em Tempo Discreto*. 2002.
- [31] G. Welch e G. Bishop. "An introduction to the kalman filter". Em: *University of North Carolina: Chapel Hill, North Carolina, US* (2006).
- [32] R. G. Brown e P. Y. Hwang. *Introduction to random signals and applied Kalman filtering: with MATLAB exercises and solutions*. John Wiley & Sons, 1997.
- [33] *Arduino - open-source tool for electronic projects*. <http://www.arduino.org>. Verificação final: Março 2016.

- [34] Omron Yaskawa Motion Control. *V1000 Quick Start Guide*. <http://www.prosoft.ru/cms/f/360949.pdf>. Verificação final: Março 2016.

Apêndice A

Tarefas Utilizadas Quando a Posição Exata da Cabina Não é Conhecida

Tarefa 1 - Interpretar

1	IF(%I1.3)THEN	25	END_IF;	53	SET %M1;
2		26		54	END_IF;
3	(*GoTo 0*)	27	(*fim—de—curso ativo*)	55	IF(NOT %I1.5)THEN
4	IF(%I1.0 AND NOT %M0	28	IF(NOT %I1.4 OR NOT %I1	56	%MW1:=5;
	AND %I1.4 AND %I1.5)		.5)THEN	57	%MW10:=%MW13;
	THEN	29	IF(NOT %M5)THEN	58	RESET %Q2.0;
5	IF(%MW0<>1)THEN	30	START %TM1;	59	SET %Q2.1;
6	%MW1:=1;	31	RESET %M1;	60	SET %Q2.2;
7	SET %M0;	32	RESET %M2;	61	SET %Q2.3;
8	END_IF;	33	RESET %M3;	62	SET %M0;
9	END_IF;	34	SET %M4;	63	SET %M1;
10		35	SET %M5;	64	END_IF;
11	(*GoTo 1*)	36	%QW5.4:=0;	65	END_IF;
12	IF(%I1.1 AND NOT %M0	37	RESET %Q2.0;	66	END_IF;
	AND %I1.4 AND %I1.5	38	RESET %Q2.1;	67	
	AND %MW0<>0 AND	39	RESET %Q2.2;	68	IF(%I1.4 AND %I1.5)THEN
	%MW0<>5)THEN	40	RESET %Q2.3;	69	DOWN %TM1;
13	IF(%MW0<>2)THEN	41	END_IF;	70	RESET %M5;
14	%MW1:=2;	42		71	END_IF;
15	SET %M0;	43	(* 3 sec parado no fim—de—	72	
16	END_IF;		curso*)	73	ELSE
17	END_IF;	44	IF(%TM1.V=%TM1.P AND	74	(* PARAGEM DE
18			%M5)THEN		EMERGENCIA*)
19	(*GoTo 2*)	45	IF(NOT %I1.4)THEN	75	%QW5.4:=0;
20	IF(%I1.2 AND NOT %M0	46	%MW1:=5;	76	
	AND %I1.4 AND %I1.5)	47	%MW10:=%MW13;	77	RESET %Q2.0;
	THEN	48	SET %Q2.0;	78	RESET %Q2.1;
21	IF(%MW0<>3)THEN	49	RESET %Q2.1;	79	RESET %Q2.2;
22	%MW1:=3;	50	SET %Q2.2;	80	RESET %Q2.3;
23	SET %M0;	51	SET %Q2.3;	81	
24	END_IF;	52	SET %M0;		

APÊNDICE A. TAREFAS UTILIZADAS QUANDO A POSIÇÃO EXATA DA CABINA NÃO É CONHECIDA

82	RESET %M0;(*botao seleccionado*)	93		103	%MW12:=4000;(*velocidade 2 patamar*)
83	RESET %M1;(*interpretar botao seleccionado*)	94	%MW6:=0;(*indicador de subida*)	104	%MW13:=2500;(*velocidade aproximacao*)
84	RESET %M2;(*perfil*)	95	%MW60:=0;(*indicador de subida auxiliar*)	105	%MW14:=5;(*velocidade incremental*)
85	RESET %M3;(*controlador PI -velocidade*)	96	%MW7:=0;(*indicador de descida*)	106	%MW15:=700;(*velocidade minima*)
86	SET %M4;(*reset de variaveis *)	97	%MW70:=0;(*indicador de descida auxiliar*)	107	%MW16:=5000;(*velocidade maxima*)
87	RESET %M5;(*temporizador ligado*)	98	%MW8:=0;(*erro PI - velocidade*)	108	%MW17:=0;(*velocidade para incrementar*)
88	RESET %M6;(*utilizado no perfil: rampa*)	99	%MW9:=0;(*velocidade PI - velocidade*)	109	
89	RESET %M7;(*utilizado no perfil: rampa degrau*)	100		110	DOWN %TM1; (* temporizador*)
90		101	%MW10:=0;(*velocidade*)	111	
91	%MW0:=0;(*pisso actual*)	102	%MW11:=3500;(*velocidade 1 patamar*)	112	END_IF;
92	%MW1:=0;(*pisso destino*)				

Tarefa 2 - Gerador

1	(* botao foi escolhido e NAO foi gerada velocidade de referencia*)	24	(* esta no piso*)	50	IF(%MW0=3)THEN RESET %M0;END_IF;
2	IF(%M0 AND NOT %M1) THEN	25	IF(%MW0=2)THEN RESET %M0;END_IF;	51	(* vai subir *)
3		26	(* vai subir *)	52	RESET %Q2.0;
4	(* GoTo 0*)	27	IF(%MW0=1)THEN	53	SET %Q2.1;
5	IF(%MW1=1)THEN	28	RESET %Q2.0;	54	SET %Q2.2;
6	(* esta no piso*)	29	SET %Q2.1;	55	SET %Q2.3;
7	IF(%MW0=1)THEN RESET %M0;END_IF;	30	SET %Q2.2;	56	IF(%MW0=1)THEN
8	(* vai descer *)	31	SET %Q2.3;	57	%MW10:=%MW12;
9	SET %Q2.0;	32	%MW10:=%MW11;	58	ELSE
10	RESET %Q2.1;	33	%MW17:=%MW10;	59	%MW10:=%MW11;
11	SET %Q2.2;	34	SET %M1;	60	END_IF;
12	SET %Q2.3;	35	END_IF;	61	%MW17:=%MW10;
13	IF(%MW0=3)THEN	36	(* vai descer *)	62	SET %M1;
14	%MW10:=%MW12;	37	IF(%MW0=3)THEN	63	END_IF;
15	ELSE	38	SET %Q2.0;	64	
16	%MW10:=%MW11;	39	RESET %Q2.1;	65	(* inicializacao *)
17	END_IF;	40	SET %Q2.2;	66	ELSIF(%M0 AND NOT %M1 AND %MW0=0)THEN
18	%MW17:=%MW10;	41	SET %Q2.3;	67	SET %Q2.0;
19	SET %M1;	42	%MW10:=%MW11;	68	RESET %Q2.1;
20	END_IF;	43	SET %M1;	69	SET %Q2.2;
21		44	END_IF;	70	SET %Q2.3;
22	(* GoTo 1*)	45	END_IF;	71	%MW10:=%MW11;
23	IF(%MW1=2)THEN	46		72	%MW17:=%MW10;
		47	(* GoTo 2*)	73	SET %M1;
		48	IF(%MW1=3)THEN	74	END_IF;
		49	(* esta no piso*)		

Tarefa 3 - Mover

```
1 (* fica no patamar *)
2 IF(%M0 AND %M1)THEN
3   IF(%MW0=%MW1)THEN
4     SET %M4;
5   (* subir*)
6   ELSIF(%Q2.1)THEN
7     IF(%MW1=2)THEN
8       IF FE %I1.6 THEN %MW6:=%MW6+1;
9       END_IF;
10    END_IF;
11    IF((%I1.6 AND %I1.7 AND NOT %I1.9)OR
12      NOT %I1.3 OR(%MW1=5 AND %I1.8)OR
13      (%I1.6 AND %I1.7 AND %MW6=2))THEN
14      SET %M4;
15    ELSE
16      SET %M2;
17    END_IF;
18  ELSE
19    SET %M2;
20  END_IF;
21  END_IF;
22  END_IF;
23  END_IF;
24  END_IF;
25  END_IF;
26  END_IF;
```

Tarefa 4 - Perfil Básico

```
1 IF(%M2)THEN
2   %MW17:=%MW10;
3   (*SET %M3;*) (*com controlador*)
4   %QW5.4:=%MW17; (*sem controlador*)
5   END_IF;
```

Tarefa 4 - Perfil Básico com Velocidade de Aproximação

```
1 IF(%M2)THEN
2   (* subir*)
3   IF(%Q2.1)THEN
4     IF FE %I1.6 THEN %MW60:=%MW60+1;
5     END_IF;
6     IF(%MW1=3 AND %MW60=4 AND %I1.6)
7     THEN
8       %MW17:=%MW13;
9     END_IF;
10    IF(%MW1=2 AND %MW60=2 AND %I1.6)
11    THEN
12      %MW17:=%MW11;
13    END_IF;
14  (* descer*)
15  ELSIF(%Q2.0)THEN
16    IF FE %I1.7 THEN %MW70:=%MW70+1;
17    END_IF;
18    IF(%MW1=1 AND %MW70=4 AND %I1.7)
19    THEN
20      %MW17:=%MW13;
21    END_IF;
22    IF(%MW1=2 AND %MW70=2 AND %I1.7)
23    THEN
24      %MW17:=%MW13;
25    END_IF;
26    IF(%MW0=0)THEN
27      %MW17:=%MW11;
28    END_IF;
29    SET %M3; (*com controlador*)
30    (* %QW5.4:=%MW17;*) (*sem controlador*)
31  END_IF;
```

Tarefa 4 - Perfil em Rampa

APÊNDICE A. TAREFAS UTILIZADAS QUANDO A POSIÇÃO EXATA DA CABINA NÃO É CONHECIDA

<pre> 1 IF(%M2)THEN 2 (* subir*) 3 IF(%Q2.1)THEN 4 IF FE %I1.6 THEN %MW60:=%MW60+1; 5 END_IF; 6 IF(%QW5.4<%MW10 AND NOT %M6)THEN 7 %MW17:=(REAL_TO_INT(INT_TO_REAL(% 8 MW10/1000)*4.0))+%QW5.4; 9 END_IF; 10 IF(%MW60=4 AND %I1.6 OR(%MW60=2 AND 11 %MW10=%MW11 AND %I1.6))THEN 12 SET %M6; 13 END_IF; 14 IF(%M6 AND %QW5.4>%MW15)THEN 15 %MW17:=%QW5.4-(REAL_TO_INT(16 INT_TO_REAL(%MW10/1000)*4.0)); 17 END_IF; 18 (* descer*) 19 ELSIF(%Q2.0)THEN </pre>	<pre> 16 IF FE %I1.7 THEN %MW70:=%MW70+1; 17 END_IF; 18 IF(%QW5.4<%MW10 AND NOT %M6)THEN 19 %MW17:=(REAL_TO_INT(INT_TO_REAL(% 20 MW10/1000)*4.0))+%QW5.4; 21 END_IF; 22 IF(%MW70=4 AND %I1.7 OR(%MW70=2 AND 23 %MW10=%MW11 AND %I1.7))THEN 24 SET %M6; 25 END_IF; 26 IF(%M6 AND %QW5.4>%MW15)THEN 27 %MW17:=%QW5.4-(REAL_TO_INT(28 INT_TO_REAL(%MW10/1000)*4.0)); 29 END_IF; 30 (* SET %M3;*) (*com controlador*) 31 %QW5.4:=%MW17; (*sem controlador*) 32 END_IF; </pre>
--	--

Tarefa 4 - Perfil em Rampa com Velocidade de Aproximação

<pre> 1 IF(%M2)THEN 2 (* subir*) 3 IF(%Q2.1)THEN 4 IF FE %I1.6 THEN %MW60:=%MW60+1; 5 END_IF; 6 IF(%QW5.4<%MW10 AND %MW60<2)THEN 7 %MW17:=(REAL_TO_INT(INT_TO_REAL(% 8 MW10/1000)*4.0))+%QW5.4; 9 END_IF; 10 IF(%MW60=4 AND %I1.7 OR(%MW60=2 AND 11 %MW10=%MW11 AND %I1.7))THEN 12 SET %M6; 13 END_IF; 14 IF(%M6 AND %QW5.4>%MW13)THEN 15 %MW17:=%QW5.4-(REAL_TO_INT(16 INT_TO_REAL(%MW10/1000)*5.5)); 17 END_IF; 18 IF(%M6 AND %MW60=4 AND %I1.6 OR(%M6 19 AND %MW60=2 AND %MW10=%MW11 20 AND %I1.6))THEN 21 SET %M7; 22 RESET %M6; 23 END_IF; 24 IF(%M7 AND %QW5.4>%MW15)THEN 25 %MW17:=%QW5.4-(REAL_TO_INT(26 INT_TO_REAL(%MW10/1000)*2.0)); 27 END_IF; 28 (* descer*) 29 ELSIF(%Q2.0)THEN </pre>	<pre> 23 IF FE %I1.7 THEN %MW70:=%MW70+1; 24 END_IF; 25 IF(%QW5.4<%MW10 AND %MW70<2)THEN 26 %MW17:=(REAL_TO_INT(INT_TO_REAL(% 27 MW10/1000)*4.0))+%QW5.4; 28 END_IF; 29 IF(%MW70=4 AND %I1.6 OR(%MW70=2 AND 30 %MW10=%MW11 AND %I1.6))THEN 31 SET %M6; 32 END_IF; 33 IF(%M6 AND %QW5.4>%MW13)THEN 34 %MW17:=%QW5.4-(REAL_TO_INT(35 INT_TO_REAL(%MW10/1000)*5.5)); 36 END_IF; 37 IF(%MW70=4 AND %I1.7 AND %M6 OR(%M6 38 AND %MW70=4 AND %MW10=%MW11 39 AND %I1.7))THEN 40 SET %M7; 41 RESET %M6; 42 END_IF; 43 IF(%M7 AND %QW5.4>%MW15)THEN 44 %MW17:=%QW5.4-(REAL_TO_INT(45 INT_TO_REAL(%MW10/1000)*2.0)); 46 END_IF; 47 SET %M3; (*com controlador*) 48 (* %QW5.4:=%MW17;*) (*sem controlador*) 49 END_IF; </pre>
--	--

Apêndice B

Tarefas Utilizadas Quando a Posição Exata da Cabina é Conhecida

Tarefa 1 - Interpretar

1	IF(%I1.3)THEN	23	%MW78:=%MW84;	51	SET %M0;
2		24	SET %M0;	52	SET %M1;
3	(*GoTo 0*)	25	END_IF;	53	END_IF;
4	IF(%I1.0 AND NOT %M0	26		54	IF(NOT %I1.5)THEN
	AND %I1.4 AND %I1.5)	27	(* fim – de – curso ativo*)	55	%MW86:=9;
	THEN	28	IF(NOT %I1.4 OR NOT %I1	56	RESET %Q2.0;
5	%MW86:=9;		.5)THEN	57	SET %Q2.1;
6	%MW79:=ABS(%MW86 – %	29	IF(NOT %M5)THEN	58	SET %Q2.2;
	MW84);	30	START %TM1;	59	SET %Q2.3;
7	%MW78:=%MW84;	31	RESET %M1;	60	SET %M0;
8	SET %M0;	32	RESET %M2;	61	SET %M1;
9	END_IF;	33	RESET %M3;	62	END_IF;
10		34	SET %M4;	63	END_IF;
11	(*GoTo 1*)	35	SET %M5;	64	END_IF;
12	IF(%I1.1 AND NOT %M0	36	%QW5.4:=0;	65	
	AND %I1.4 AND %I1.5)	37	RESET %Q2.0;	66	IF(%I1.4 AND %I1.5)THEN
	THEN	38	RESET %Q2.1;	67	DOWN %TM1;
13	%MW86:=63;	39	RESET %Q2.2;	68	RESET %M5;
14	%MW79:=ABS(%MW86 – %	40	RESET %Q2.3;	69	END_IF;
	MW84);	41	END_IF;	70	
15	%MW78:=%MW84;	42		71	ELSE
16	SET %M0;	43	(* 3 sec parado no fim – de –	72	(* PARAGEM DE
17	END_IF;		curso*)		EMERGENCIA*)
18		44	IF(%TM1.V=%TM1.P AND	73	%QW5.4:=0;
19	(*GoTo 2*)		%M5)THEN	74	
20	IF(%I1.2 AND NOT %M0	45	IF(NOT %I1.4)THEN	75	RESET %Q2.0;
	AND %I1.4 AND %I1.5)	46	%MW86:=117;	76	RESET %Q2.1;
	THEN	47	SET %Q2.0;	77	RESET %Q2.2;
21	%MW86:=117;	48	RESET %Q2.1;	78	RESET %Q2.3;
22	%MW79:=ABS(%MW86 – %	49	SET %Q2.2;	79	
	MW84);	50	SET %Q2.3;		

APÊNDICE B. TAREFAS UTILIZADAS QUANDO A POSIÇÃO EXATA DA CABINA É CONHECIDA

80	RESET %M0;(*botao	90	%MW0:=0;(*pisos actual*)	102	%MW12:=4000;(*velocidade 2
	selecionado*)	91	%MW1:=0;(*pisos destino*)		pisos*)
81	RESET %M1;(*interpretar	92		103	%MW13:=2500;(*velocidade
	botao selecionado*)	93	%MW6:=0;(*indicador de		aproximacao*)
82	RESET %M2;(*perfil*)		subida*)	104	%MW14:=5;(*velocidade
83	RESET %M3;(*controlador PI	94	%MW60:=0;(*indicador de		incremental*)
	— velocidade*)		subida*)	105	%MW15:=700;(*velocidade
84	SET %M4;(*reset PI—	95	%MW7:=0;(*indicador de		min*)
	velocidade*)		descida*)	106	%MW16:=5000;(*velocidade
85	RESET %M5;(*temporizador	96	%MW70:=0;(*indicador de		max*)
	ligado*)		descida*)	107	%MW17:=0;(*velocidade para
86	RESET %M6;(*utilizado no	97	%MW8:=0;(*erro PI —		incrementar*)
	perfil: rampa*)		velocidade*)	108	
87	RESET %M7;(*utilizado no	98	%MW9:=0;(*velocidade PI —	109	DOWN %TM1; (*
	perfil: rampa degrau*)		velocidade*)		temporizador*)
88	RESET %M8;(*utilizado no	99		110	
	perfil: funcao da posicao	100	%MW10:=0;(*velocidade*)	111	END_IF;
	*)	101	%MW11:=3500;(*velocidade 1		
89			pisos*)		

Tarefa 2 - Gerador

1	IF(%M0 AND NOT %M1)	10	SET %Q2.2;	23	RESET %Q2.1;
	THEN	11	SET %Q2.3;	24	SET %Q2.0;
2	(* esta no piso*)	12	%MW78:=%MW78—6;	25	SET %Q2.2;
3	IF(%MW84=%MW86 OR %	13	IF(%MW79>70)THEN	26	SET %Q2.3;
	MW84+1=%MW86 OR	14	%MW10:=%MW12;	27	%MW78:=%MW78+6;
	%MW84—1=%MW86)	15	ELSE	28	IF(%MW79>70)THEN
	THEN	16	%MW10:=%MW11;	29	%MW10:=%MW12;
4	RESET %M0;	17	END_IF;	30	ELSE
5	END_IF;	18	%MW17:=%MW10;	31	%MW10:=%MW11;
6	(* vai subir *)	19	SET %M1;	32	END_IF;
7	IF(%MW86>%MW84)THEN	20	END_IF;	33	SET %M1;
8	SET %Q2.1;	21	(* vai descer *)	34	END_IF;
9	RESET %Q2.0;	22	IF(%MW86<%MW84)THEN	35	END_IF;

Tarefa 3 - Mover

1	IF(%M0 AND %M1)THEN	6	ELSE
2	(* para*)	7	SET %M2;
3	IF(%MW86=%MW84)THEN	8	END_IF;
4	SET %M4;	9	END_IF;
5	(* move—se*)		

Tarefa 4 - Perfil de Velocidade em Tarefa da Posição

<pre> 1 IF(%M2)THEN 2 (* inicio do movimento*) 3 IF(NOT %M8)THEN 4 (* contrloador PI*) 5 %MW88:=ABS(%MW78-%MW84); 6 %MW80:=(160*%MW88)+%MW90; 7 %MW90:=%MW90+REAL_TO_INT(8 INT_TO_REAL(%MW88)*0.5); 9 IF(%MW80<%MW15)THEN %MW80:=% 10 MW15;END_IF; 11 IF(%MW80>%MW10)THEN 12 %MW80:=%MW10; 13 SET %M8; 14 %MW90:=0; 15 END_IF; 16 %MW17:=%MW80; 17 END_IF; </pre>	<pre> 17 (* inicio de paragem*) 18 IF(%M8)THEN 19 (* contrloador PI*) 20 %MW88:=ABS(%MW86-%MW84); 21 %MW80:=(150*%MW88)+%MW90; 22 %MW90:=%MW90+REAL_TO_INT(23 INT_TO_REAL(%MW88)*0.02); 24 IF(%MW80<%MW15)THEN %MW80:=% 25 MW15;END_IF; 26 IF(%MW80>%MW10)THEN %MW80:=% 27 MW10;END_IF; 28 %MW17:=%MW80; 29 END_IF; 30 (*SET %M3;*) (*com controlador*) 31 %QW5.4:=%MW17; (*sem controlador*) 32 END_IF; </pre>
--	--

Tarefa 4 - Perfil com Limitação de Aceleração

<pre> 1 IF(%M2)THEN 2 %MW300:=ABS(%MW78-%MW84); 3 %MW301:=ABS(%MW86-%MW84); 4 %MF200:=(INT_TO_REAL(%IW5.3)-300.0) 5 /(3600.0-300.0); 6 (* supervisor*) 7 IF(%MW300<15 AND %MW301>15)THEN % 8 MF202:=0.55-%MF200;END_IF; 9 IF(%MW300>=15 AND %MW301>=15)THEN % 10 MF202:=0.505-%MF200;END_IF; 11 IF(%MW300>15 AND %MW301<15)THEN % 12 MF202:=0.45-%MF200;END_IF; 13 (* controlador PI*) </pre>	<pre> 10 %MF204:=%MF202*%MF208+%MF206;(*ganho 11 proporcional*) 12 %MF206:=%MF206+%MF202*%MF210;(*saida*) 13 %MW199:=REAL_TO_INT(%MF204); 14 (* saturacao*) 15 IF(%MW199>%MW16)THEN %MW199:=% 16 MW16;END_IF; 17 IF(%MW199<%MW15)THEN %MW199:=% 18 MW15;END_IF; 19 %MW17:=%MW199; 20 SET %M3; (*com controlador*) 21 (* %QW5.4:=%MW17;*) (*sem controlador*) 22 END_IF; </pre>
---	--

Apêndice C

Tarefas Adicionais

Controlador

```
1 IF(%M3)THEN
2   %MF116:=(INT_TO_REAL(%MW16)/10000.0);
3   %MF118:=(INT_TO_REAL(%MW15)/10000.0);
4   (* erro*)
5   %MF112:=(INT_TO_REAL((%MW17-%IW5.0))
6     /10000.0);
7   (* ganho proporcional*)
8   %MF100:=%MF102*%MF112;
9   (* saida*)
10  %MF114:=%MF100+%MF104;
11  (* saturacao*)
12  IF(%MF114>%MF116)THEN %MF114:=%MF116;
13    END_IF;
14
15  IF(%MF114<%MF118)THEN %MF114:=%MF118;
16    END_IF;
17
18  %MW120:=REAL_TO_INT(%MF114*10000.0);
19  %QW5.4:=%MW120;
20
21  (* ganho integral*)
22  %MF106:=%MF104+%MF108*%MF112;(*+%
23    MF110*((INT_TO_REAL(%QW5.4)/
24    INT_TO_REAL(%MW16))- %MF114);*)
25  %MF104:=%MF106;
26  END_IF;
```

Reset de Parcial

```
1 IF(%M4)THEN
2   %MF100:=0.0;(* P *)
3   %MF102:=0.9;(* Kp
4     *)
5   %MF104:=0.0;(* I(k
6     - 1)*)
7   %MF106:=0.0;(* I(k
8     *)
9   %MF108:=%MF102
10  *0.1/2.25;(*bi=
11  Kp*Ts/Ti*)
12
13  %MF110:=0.1/0.01;
14  (*ao=Ts/Tt*)
15  %MF112:=0.0;(*erro
16  *)
17  %MF114:=0.0;(*v*)
18
19  %MF206:=0.0;
20  %MF208:=5.9;
21  %MF210:=%MF208
22  *0.1/0.008;
23
24  %QW5.4:=0;
25  RESET %Q2.0;
26  RESET %Q2.1;
27  RESET %Q2.2;
28  RESET %Q2.3;
29  RESET %M0;
30  RESET %M1;
31  RESET %M2;
32  RESET %M3;
33  RESET %M5;
34  RESET %M6;
35  RESET %M7;
36
37  RESET %M8;
38  %MW0:=%MW1;
39  %MW6:=0;
40  %MW60:=0;
41  %MW7:=0;
42  %MW70:=0;
43  %MW90:=0;
44  %MW17:=0;
45  %QW5.4:=0;
46  RESET %M4;
47  END_IF;
```

Interpretador do sensor de Ultra Sons

```

1  %MF80:=INT_TO_REAL(%IW5.1); (*ultra sons, float*)
2  %MF84:=0.0339752*%MF80-9.689; (*pos actual, float*)
3  %mW84:=REAL_TO_INT(%MF84); (*pos actual, int*)
4  %MF86:=INT_TO_REAL(%MW86); (*pos destino, float*)

```

Apêndice D

Código Arduino

Teste das Seis Posições Fixas com o MPU6050

```
1 #include "I2Cdev.h"          19 //reset de offsets          35 /* Descomentar o eixo que
2 #include "MPU6050.h"        20 mpu.setXAccelOffset(0);      se quer estudar */
3 #include "Wire.h"           21 mpu.setYAccelOffset(0);      36 acc=mpu.getAccelerationX();
4                               22 mpu.setZAccelOffset(0);
5 //para calibrar: [10 ; 15]   23                               37 // acc=mpu.
6 #define TEMPO 15            24 //defin escala: +-4G        getAccelerationY();
7                               25 mpu.setFullScaleAccelRange( 38 // acc=mpu.
8 MPU6050 mpu;                26 }                             getAccelerationZ();
9                               27                               39 Serial.println(acc);
10 int16_t acc;                28                               40
11                               29 void loop() {                41 //garantir mesma
12 void setup() {               30 //converso para            frequencia de aquisicao
13 //inicializar porta serie    31 while ( millis () <(TEMPO   time=millis() -time;
14 Serial.begin(9600);          32 *1000*60)){                 if (time < 10)
15                               33 long time = millis () ;     delay(10-time);
16 //inicializar acelerometro   34                               45 }
17 mpu.initialize() ;          46                               46 }
18
```

Estudo da Influência da Temperatura nas Medições do MPU6050

```
1 #include<MPU6050.h>          11 double accX=0, accY=0, accZ  21 //reset de offsets
2 #include <SD.h>               12 =0;                          22 mpu.setXAccelOffset(0);
3 #include <Wire.h>             13                               23 mpu.setYAccelOffset(0);
4 #include <LCD03.h>            14 void setup() {               24 mpu.setZAccelOffset(0);
5                               15 //inicializar lcd           25
6 LCD03 lcd;                   16 lcd.begin(16,2);            26 //definir escala: +-4G
7 MPU6050 mpu;                 17 lcd.backlight();            27 mpu.setFullScaleAccelRange(
8 File myFile;                 18                               MPU6050_ACCEL_FS_4);
9                               19 //inicializar acelerometro  28
10 int16_t acX=0, acY=0, acZ=0; 19 mpu.initialize() ;          29 //inicializar cartao SD
                               20                               30 pinMode(10, OUTPUT);
```

```

31 if (SD.begin(4)) {
32   myFile = SD.open("
      efeitoTemp.txt",
      FILE_WRITE);
33   myFile.println("\n Inicio de
      sessao");
34   myFile.close();
35   delay(1);
36 }
37 }
38
39 void loop(){
40   long time = millis ();
41   // conversao para graus
      celsius
42   double temp = (mpu.
      getTemperature()/340.0) +
      36.53;
43
44   mpu.getAcceleration(&acX, &
      acY, &acZ);
45
46   //converso para m/s2
      utilizando os resultados da
      calibracao pelo metodo fixo
      das 6 posicoes
47   accX = ((double)acX
      -7844.269926)/ 834.458849;
48   accY = ((double)acY
      +4335.217684)/837.70399;
49   accZ = ((double)acZ
      +3955.968)/855.7013;
50
51   //gravar no cartao SD:
      tempo - temperatura -
      acc x - acc y - acc z
52   myFile = SD.open("
      efeitoTemp.txt",
      FILE_WRITE);
53   myFile.print(millis () );
54   myFile.print("\t");
55   myFile.print(temp,1);
56   myFile.print("\t");
57   myFile.print(accX,4);
58   myFile.print("\t");
59   myFile.print(accY,4);
60   myFile.print("\t");
61   myFile.println(accZ,4);
62   myFile.close();
63   delay(2);
64
65   //mostrar a temperatura no
      LCD para saber quando
      parar de recolher dados
66   lcd.clear();
67   lcd.print("Temp:");
68   lcd.print(temp,2);
69
70   //garantir mesma frequencia
      de aquisicao
71   time=millis()-time;
72   if (time < 50)
73     delay(50-time);
74 }

```

Utilização do Filtro de Kalman nas Medições do MPU6050

```

1  #include "I2Cdev.h"
2  #include <Wire.h>
3  #include<MPU6050.h>
4
5  #define WITHE_NOISE_PSD
      0.1
6  #define DT 70/1000
7
8  MPU6050 mpu; // uint8_t
      MPU=0x68;
9
10 int16_t az;
11
12 //inicializar matrizes para o
      filtro de Kalman
13 double xHatMinus
      [3][1]={0},{0},{0};
14 double xHat[3][1]={0},{0},{0};
15 double P[3][3]={10, 0, 0},{0,
      10, 0},{0, 0, 10};
16 double Pminus[3][3]={0, 0,
      0},{0, 0, 0},{0, 0, 0};
17 double K[3][2]={0, 0},{0,
      0},{0, 0};
18
19 //definir variaveis para
      medir tempo
20 long time;
21 long time2;
22
23
24 void setup(){
25   Serial.begin(9600);
26
27   mpu.initialize();
28   mpu.setZAccelOffset(0);
29   mpu.setFullScaleAccelRange(
      MPU6050_ACCEL_FS_4);
30
31   Serial.println("Acc \t Vel \t
      Pos");
32 }
33
34
35 void loop(){
36   while (millis () < 300000){ //
      duracao de 5 min ( 5 min *
      60 sec * 1000
      millisegundos)
37   time = millis ();
38
39   //valor da aceleracao (raw)
40   az = mpu.getAccelerationZ()
      ;
41
42   //aceleracao em m/s2
43   double AZ = ((double)az
      +3955.968)/855.7013;
44
45   //aplicao do filtro de
      Kalman para estimar
      posicao e velocidade e
      filtrar aceleracao
46   filtrado (AZ);
47
48   //manter uma frequencia
      constante
49   time2 = millis () -time;
50   if (time2 <150) {
51     delay(150-time2);
52   }
53 }
54
55
56
57 void filtrado ( double
      aceleracaoZ){

```

58	// project the stat ahead	70	Pminus[1][1]= constrain(P	87	xHat[0][0]=xHatMinus[0][0]+
59	xHatMinus[0][0]=xHat[0][0]+		[2][2]* DT*DT+P[1][2]*DT+		K[0][0]*(aceleracaoZ*factor
	xHat[1][0]*DT+xHat[2][0]*		P[2][1]*DT+P[1][1]+		-xHatMinus[2][0]);
	DT*DT/2; //acc		WITHE_NOISE_PSD * DT*	88	xHat[1][0]=xHatMinus[1][0]+
60	xHatMinus[1][0]=xHat[1][0]+		DT*DT/3, -1.0e9, 1.0e9);		K[1][0]*(aceleracaoZ*factor
	xHat[2][0]*DT; //vel	71	Pminus[2][1]= constrain(P		-xHatMinus[2][0]);
61	xHatMinus[2][0]=xHat[2][0];		[2][1]+P[2][2]* DT+	89	xHat[2][0]=xHatMinus[2][0]+
	//pos		WITHE_NOISE_PSD * DT*		K[2][0]*(aceleracaoZ*factor
62			DT/2, -1.0e9, 1.0e9);		-xHatMinus[2][0]);
63	// project the error	72		90	
	covariance ahead	73	Pminus[0][2]= constrain((P	91	//update error covariance
64	// Pminus= A * P * A^T + Q		[2][2]* DT*DT/2+P[1][2]*	92	//P = (I - K * C) * Pminus
65	Pminus[0][0]= constrain((P		DT+2*P[0][2])+	93	P[0][0]= Pminus[0][0]-
	[2][2]* DT*DT*DT*DT+2*P		WITHE_NOISE_PSD * DT*		Pminus[2][0]*K[0][0];
	[1][2]*DT*DT*DT+2*P		DT*DT/6, -1.0e9, 1.0e9);	94	P[1][0]= Pminus[1][0]-
	[2][1]*DT*DT*DT+2*P	74	Pminus[1][2]= constrain(P		Pminus[2][0]*K[1][0];
	[0][2]*DT*DT+4*P[1][1]*DT		[1][2]+P[2][2]* DT+	95	P[2][0]= Pminus[2][0]*(1-K
	*DT+2*P[2][0]*DT*DT+4*P		WITHE_NOISE_PSD * DT*		[2][0]);
	[0][1]*DT+4*P[0][0])/4 +		DT/2, -1.0e9, 1.0e9);	96	
	WITHE_NOISE_PSD * DT*	75	Pminus[2][2]= constrain(P	97	P[0][1]= Pminus[0][1]-
	DT*DT*DT*DT/20, -1.0e9,		[2][2]+		Pminus[2][1]*K[0][0];
	1.0e9);		WITHE_NOISE_PSD * DT,	98	P[1][1]= Pminus[1][1]-
66	Pminus[1][0]= constrain(2*(P		-1.0e9, 1.0e9);		Pminus[2][1]*K[1][0];
	[2][2]* DT*DT*DT+P[1][2]*	76		99	P[2][1]= Pminus[2][1]*(1-K
	DT*DT+2*P[2][1]*DT*DT	77	//Compute kalman gain		[2][0]);
	+2*P[1][1]*DT+2*P[2][0]*	78	// K = Pminus * C^T * (C *	100	
	DT+2*P[1][0])/4 +		Pminus * C^T + R)^-1	101	P[0][2]= Pminus[0][2]-
	WITHE_NOISE_PSD * DT*	79	K[0][0]= (Pminus[0][2]/(Pminus[2][2]*K[0][0];
	DT*DT*DT/8, -1.0e9, 1.0		Pminus[2][2]+0.004));	102	P[1][2]= Pminus[1][2]-
	e9);	80	K[1][0]= (Pminus[1][2]/(Pminus[2][2]*K[1][0];
67	Pminus[2][0]= constrain(2*(P		Pminus[2][2]+0.004));	103	P[2][2]= Pminus[2][2]*(1-K
	[2][2]* DT*DT+2*P[2][1]*	81	K[2][0]= (Pminus[2][2]/([2][0]);
	DT+2*P[2][0])/4 +		Pminus[2][2]+0.004));	104	
	WITHE_NOISE_PSD * DT*	82		105	//imprimir resultados
	DT*DT/6, -1.0e9, 1.0e9);	83	float factor=1.3; //		estimados
68			filtragem muito intensa!	106	Serial.print(xHat[2][0],4) ;
69	Pminus[0][1]= constrain(((P		Necessidade de compensar		//acc
	[2][2]* DT*DT*DT+2*P		o resultado filtrado para	107	Serial.print("\t");
	[1][2]*DT*DT+P[2][1]*DT*		ter um valor mais proximo	108	Serial.print(xHat[1][0],2) ;
	DT+2*P[0][2]*DT+2*P		do real		//vel
	[1][1]*DT+2*P[0][1])/2+	84		109	Serial.print("\t");
	WITHE_NOISE_PSD * DT*	85	//Update estimate with	110	Serial.println(xHat[0][0],2) ;
	DT*DT*DT/8, -1.0e9, 1.0		measurement		//pos
	e9);	86	// xHat = xHatMinus + K (z	111	}
			- C * xHatMinus)		

Utilização do Filtro de Kalman nas Medições do BMP180

1	#include "I2Cdev.h"	4		7	
2	#include <Wire.h>	5	double calibrarBMP(void);	8	BMP085 bmp;
3	#include <BMP085.h>	6	void filtro (double posicao);	9	

```

10 double xHat= 0.0;
11 double xHatMinus=0.0;
12 double P=10.0;
13 double Pminus=0.0;
14 double K=0.0;
15 double R=0.5;
16 double Q=1e-3;
17
18 double pressaoReferencia;
19 long time;
20 long time2;
21
22 void setup(){
23   Serial.begin(9600);
24
25   bmp.begin(
26     BMP085_ULTRA_HIGH_RES
27   );
28   //fazer o zero do BMP
29   pressaoReferencia = (double)
30     calibrarBMP();
31   Serial.println("Sem filtro \t
32     Com filtro");
33 }
34
35 void loop(){
36   while ( millis () < 300000){
37     time = millis ();
38     double pressao = (double)
39       bmp.readPressure();
40     double posicao = bmp.
41       getAltitude(pressao,
42         pressaoReferencia);
43     filtro ( posicao);
44
45     time2 = millis () –time;
46     if (time2 <150) {
47       delay(150–time2);
48     }
49   }
50
51   double calibrarBMP(void){
52     double pressau;
53     double xHatMinusBMP,
54       xHatBMP=101325,
55       PminusBMP= 0, PBMP =
56       1000, KBMP;
57     double QBMP = 1e–3, RBMP
58       = 0.5;
59     xHatBMP = (double) bmp.
60       readPressure();
61     for ( int i = 0; i<200; i++){
62       xHatMinusBMP=xHatBMP;
63       PminusBMP= PBMP+QBMP
64       ;
65       KBMP = PminusBMP/(
66         PminusBMP+RBMP);
67       xHatBMP= xHatMinusBMP
68         + KBMP * ((double) bmp.
69           readPressure() –
70           xHatMinusBMP);
71       PBMP = (1–KBMP)*
72         PminusBMP;
73     };
74     return (double) xHatBMP;
75   }
76 }
77
78 void filtro (double posicao){
79   // project the stat ahead
80   xHatMinus=xHat;
81
82   // project the error
83   covariance ahead
84   // Pminus= A * P * A^T + Q
85   Pminus= P+Q;
86
87   //Compute kalman gain
88   // K = Pminus * C^T * (C *
89     Pminus * C^T + R )^–1
90   K = Pminus/(Pminus+R);
91
92   //Update estimate with
93   measurement
94   // xHat = xHatMinus + K ( z
95     – C * xHatMinus)
96   xHat = xHatMinus + K * (
97     posicao – xHatMinus);
98
99   //update error covariance
100   //P = (I – K * C ) * Pminus
101   P = (1–K)*Pminus;
102
103   Serial.print(posicao,2);
104   Serial.print("\t");
105   Serial.println(xHat,2);
106 }

```

Utilização do Filtro de Kalman nas Medições do HC-SR04

```

1 //
2 // Example NewPing library
3 // sketch that does a ping
4 // about 20 times per second.
5
6 #include <Wire.h>
7 #include <NewPing.h>
8 #define ALIMENTACAO 2
9
10 #define TRIGGER_PIN 12 //
11 // Arduino pin tied to trigger
12 // pin on the ultrasonic sensor.
13
14 #define ECHO_PIN 11 //
15 // Arduino pin tied to echo pin
16 // on the ultrasonic sensor.
17
18 #define MAX_DISTANCE 200
19 // Maximum distance we
20 // want to ping for (in
21 // centimeters). Maximum
22 // sensor distance is rated at
23 // 400–500cm.
24
25 #define PWM_PIN 3
26 #define ANALOG_PIN 3
27
28 NewPing sonar(
29   TRIGGER_PIN, ECHO_PIN,
30   MAX_DISTANCE); //
31 // NewPing setup of pins and
32 // maximum distance.
33
34 unsigned int distancia;
35 unsigned long time;
36 int sensorValue = 0;
37 float volt = 0.0;

```

18		34	void loop() {	50	distancia = distancia * 2;
19		35	time = millis ();	51	if (distancia >255)
20	float xHat=125, Pminus=10;	36	distancia = sonar.ping_cm();		distancia =255;
21	float xHatMinus,P,K;			52	if (distancia <0) distancia
22	float Q = .8;	37			=0;
23	float R = 1;	38		53	analogWrite(PWM_PIN,
24		39	xHatMinus=xHat;		distancia); // 0 <
25		40	Pminus=P+Q;		distancia <
26	void setup() {	41	K=Pminus/(Pminus+R);		MAX_DISTANCE < 255
27	Serial.begin(9600);	42	xHat=xHatMinus+K*(54	// sensorValue =
28	pinMode(PWM_PIN,		distancia-xHatMinus);		analogRead(
	OUTPUT);	43	P=(1-K)*P;		ANALOG_PIN);
29	pinMode(ALIMENTACAO,	44	distancia = (int) xHat;	55	// volt = sensorValue *
	OUTPUT);	45			(5.0 / 1023.0);
30	delay(2);	46		56	
31	digitalWrite (47	time = millis () - time;	57	Serial.print("\n");
	ALIMENTACAO,HIGH);	48	if (time < 100) delay(100	58	}
32	}		- time);		
33		49	Serial.print(distancia);		

Software do Sistema de Medição

1	#define WITHE_NOISE_PSD	24	/* -----	36	void printBatUSB(void); //
	0.1		Prototipos		imprime simbolo que esta a
2	#define DT 70/1000		----- */		ser alimentado por USB
3	#define KEYPAD_ A0	25	// -- prototipos para LCD	37	
4	#define POWER_SUPLY_ 2	26	void startLCD(void); //	38	// -- prototipos para
5	#define LED_ 3		Inicializa LCD		TECLADO
6	#define BAT_LEVEL_ A10	27	void printLCDbasics(float lvl	39	int handleKeypad(int pin); //
7	#define PWM_PORTA_ 6); //Imprime caracteres		Intepreta o botao carregado
8			basicos para navegar nos	40	
9			menus (setas + bateria)	41	// -- prototipos para SD
10	#include <Wire.h>	28	void displayMenu(); //Mostra	42	void saveData (double accX,
11	#include <SPI.h>		sistema de menus no LCD		double accY, double pos,
12	#include <SD.h>	29			double velo, double acc); //
13	#include <BMP085.h>	30	// -- prototipos para		guardar todos os dados no
14	/* AUTHORS: Korneliusz		BATERIA		cartao SD
	Jarzebski*/	31	void battCheckUp(void); //	43	void saveDataACC (double
15	#include<MPU6050.h>		mede a tensao na bateria		accX, double accY, double
16	/* AUTHORS: Jeff Rowberg*/	32	void printBatLCD(float lvl);		accZ); // guardar apenas
17	#include <MenuSystem.h>		//imprime o simbolo do		valores de aceleracao no
18	/* AUTHORS: Guillaume S,		estado da Bateria		cartao SD
	Jon Black, Nicolas, Thomas	33	void printBatLow(void); //	44	
	Jarosch*/		imprime simbolo de bateria	45	// -- prototipos para
19	#include <LCD03.h>		fraca		MEDICAO
20	/* AUTHORS: Ben Arblaster	34	void printBat50(void); //	46	double calibrarBMP(void); //
	*/		imprime simbolo de bateria a		fazer zero ao barometro
21	#include "thread.h"		meia carga		
22	/* AUTHORS: Lino Estevao*/	35	void printBat100(void); //		
23			imprime simbolo de bateria		
			cheia		


```

47 void medicoes( double* pos,
    double * aX, double * aY,
    double * aZ, double
    pressaoRef, int modelo=0,
    double biasAccX=
    -7844.269926, double
    razaoX= 834.458849, double
    biasAccY=4335.217684,
    double razaoY=837.70399,
    double biasAccZ= 3955.968,
    double razaoZ=855.7013); //
    leitura dos sensores
48 void medicoes1( double* pos,
    double * aX, double * aY,
    double * aZ, double
    pressaoRef, int modelo=0,
    double biasX=0, double
    biasY=0, double biasZ=0,
    double biasAccX=
    -7844.269926, double
    razaoX= 834.458849, double
    biasAccY=4335.217684,
    double razaoY=837.70399,
    double biasAccZ= 3955.968,
    double razaoZ=855.7013); //
    leitura dos sensores apos
    fazer o zero
49
50 // -- prototipos para
    KALMAN
51 void resetKalman(void); //
    Reset das variaveis do filtro
    de kalman
52 void filtro (double*
    aceleracaoZ, double*
    aceleracaoY, double*
    aceleracaoX); // filtro
    aceleracao apenas
53 void filtro2 (double* posicao,
    double* velocidade, double*
    aceleracaoZ, double*
    aceleracaoY, double*
    aceleracaoX); // filtro para
    navegacao inercial
54
55
56 /* -----
    Funcoes Menu
    ----- */
57 void on_item1_selected(
    MenuItem* p_menu_item);
58 void on_item2_selected(
    MenuItem* p_menu_item);
59 void on_item3_selected(
    MenuItem* p_menu_item);
60 void on_item4_selected(
    MenuItem* p_menu_item);
61 void on_item5_selected(
    MenuItem* p_menu_item);
62 void on_item6_selected(
    MenuItem* p_menu_item);
63
64 /* -----
    MenuSystem.H
    ----- */
65 MenuSystem ms;
66 Menu mm("ROOT");
67 Menu mF("Modo Func.");
68 MenuItem mF_m1("Det. Perfis
    ");
69 MenuItem mF_m2("Avaliacao
    ");
70 MenuItem mF_m6("Aux.
    Control.");
71 Menu mFF("Config.");
72 MenuItem mF_m3("LCD luz");
73 MenuItem mF_m4("Desligar");
74 MenuItem mF_m5("Tensao Bat
    .");
75 LCD03 lcd;
76 int backLight = 1;
77
78
79
80 /* ---- definir sensores e
    variavies associadas ---- */
81 MPU6050 mpu; // uint8_t
    MPU=0x68;
82 double aceleracaoX,
    aceleracaoY,
    aceleracaoZ, posicao,
    velocidade;
83
84 BMP085 bmp; // uint8_t BMP
    =0x77;
85 double pressaoReferencia,
    pressao, pos;
86
87
88 /* ---- definir carto de
    memoria ---- */
89 File myFile;
90
91
92 /* ---- definir matrizes para
    kalman ---- */
93 double xHat_[4]={0,0,0,0};
94 double xHatMinus_
    [4]={0,0,0,0};
95 double P_[4]={10,10,10,10};
96 double Pminus_[4]={0,0,0,0};
97 double K_[4]={0,0,0,0};
98 double R_
    [4]={0.004,0.004,0.004,0.5};
99 double Q_[4]={1e-3,1e-3,1e
    -3,1e-3};
100 double xHatMinus
    [3][1]={0},{0},{0};
101 double xHat[3][1]={0},{0},{0};
102 double P
    [3][3]={0,0,0},{0,10,0},{0,0,10};
103 double Pminus
    [3][3]={0,0,0},{0,0,0},{0,0,0};
104 double K
    [3][2]={0,0},{0,0},{0,0};
105
106
107 /* ---- bateria ---- */
108 float nivel_de_bat=10.0;
109
110 thread battery(battCheckUp
    ,5000,true);
111
112
113 /* -----
    setup
    ----- */
114 void setup(){
115 //definir pin
116 pinMode(POWER_SUPPLY_,
    OUTPUT);
117 pinMode(LED_, OUTPUT);
118 pinMode(PWM_PORTA_,
    OUTPUT);
119 digitalWrite(
    POWER_SUPPLY_, HIGH);
120 digitalWrite(LED_, HIGH);
121 analogWrite(PWM_PORTA_,
    0);
122 pinMode(KEYPAD_, INPUT);
123
124 //inicializar LCD e criar
    sistema de menu
125 startLCD();

```


126	mm.add_menu(&mF); //	158	myFile = SD.open("Aceleracoes.txt", FILE_WRITE);	198	case 1:
	Modo Func.			199	lcd.clear();
127	mF.add_item(&mF_m1, &on_item1_selected); //	159	myFile.println("\n Inicio de sessao");	200	ms.back();
	Perfis			201	printLCDbasics(nivel_de_bat);
128	mF.add_item(&mF_m2, &on_item2_selected); //	160	myFile.close();	202	displayMenu();
	Avaliacao	161	delay(1);	203	break;
		162	}	204	
129	mF.add_item(&mF_m6, &on_item6_selected); //	163		205	
	PWM	164	//Apresentar coisas no LCD	206	case 4:
130	mm.add_menu(&mFF); //	165	lcd.clear();	207	lcd.clear();
	Config.	166	lcd.print("Bem vindo");	208	ms.select();
131	mFF.add_item(&mF_m3, &on_item3_selected); //	167	battCheckUp();	209	printLCDbasics(nivel_de_bat);
	LCD luz	168	delay(400);	210	displayMenu();
132	mFF.add_item(&mF_m5, &on_item5_selected); //	169	lcd.clear();	211	break;
	Nivel Bateria	170	printLCDbasics(nivel_de_bat);	212	
133	mm.add_item(&mF_m4, &on_item4_selected); //	171	;	213	case 0:
	Desligar	172	displayMenu();	214	break;
134	ms.set_root_menu(&mm); //	173	}	215	}
	ROOT	174		216	
		175	/* -----	217	//delay necessario para o correcto funcionamento
135		176	loop	218	delay(150);
136	//inicializar MPU	177	----- */	219	
137	mpu.initialize();	178	void loop(){	220	}
138	mpu.setZAccelOffset(0);	179	//ver nivel da bateria a cada 5 segundos	221	
139	mpu.setYAccelOffset(0);	180	battery.handleRun();	222	
140	mpu.setXAccelOffset(0);	181		223	/* -----
141	mpu.setFullScaleAccelRange(MPU6050_ACCEL_FS_4);	182	//ler tecla primirda		LCD – startLCD – Inicializa LCD
142		183	int opcao = handleKeypad(KEYPAD_);		----- */
143	//inicializar BMP	184		224	void startLCD(void){
144	bmp.begin(0x03);	185		225	lcd.begin(16,2);
145	/*	186	//navegar no sistema de menus conforme a tecla primida	226	lcd.noBacklight();
146	ULTRA_HIGH_RES = 0x03,	187	switch (opcao) {	227	lcd.print("Inicializando... ");
147	HIGH_RES = 0x02,	188	case 2:	228	delay(100);
148	STANDARD = 0x01,	189	lcd.clear();	229	
149	ULTRA_LOW_POWER = 0	190	ms.prev();	230	//criar caracteres especiais
150	x00	191	printLCDbasics(nivel_de_bat);	231	byte batEmpty[8] = {0b01110,0b11011,0b10001,0b10001,0b10001,0b10001,0b11111};
151	*/	192	displayMenu();	232	byte bat50[8] = {0b01110,0b11011,0b10001,0b11111,0b11111,0b11111,0b11111,0b11111};
152	//inicializar Carto e ficheiros	193	break;	233	byte bat100[8] = {0b01110,0b11111,0b11111,0b11111,0b11111,0b11111,0b11111,0b11111};
153	pinMode(10, OUTPUT);	194			
154	if (SD.begin(4)) {	195	case 3:		
155	myFile = SD.open("Perfil.txt", FILE_WRITE);	196	lcd.clear();		
156	myFile.println("\n Inicio de sessao");	197	ms.next();		
157	myFile.close();		printLCDbasics(nivel_de_bat);		
			displayMenu();		
			break;		

```

234 byte setaCima[8] = {0b00100 261 lcd.print(cp_menu->
,0b01110,0b11011,0b00000,0
b00000,0b00000,0b00000,0
b00000}; 262 } 292 if (nivel_de_bat > 6.2 &&
nivel_de_bat < 7.2){
235 byte setaBaixo[8] = {0b00000 263 lcd.clear ();
,0b00000,0b00000,0b00000,0
b00000,0b11011,0b01110,0 264 } 293 lcd.println("Bateria fraca");
236 byte setaCimaBaixo[8] = {0 265 /* -----
b00100,0b01110,0b11011,0
b00000,0b00000,0b11011,0
b01110,0b00100}; 266 LCD - printLCDbasics -
imprime caracteres basicos
para navegar nos menus (
setas + bateria) 267 } 294 lcd.println("a desligar... ");
237 byte ultimo[8] = {0b01000,0 268 /* -----
b01100,0b01110,0b01111,0
b01111,0b01110,0b01100,0 269 } 295 lcd.println("a desligar... ");
b01000}; 270 void printLCDbasics(float lvl 300 delay(500);
238 byte extra[8] = {0b00000,0 271 }{ 301 digitalWrite(
b01010,0b01010,0b11111,0 272 boolean cima = ms. 302 POWER_SUPPLY_, LOW);
b11111,0b11111,0b01110,0 273 theresNext(); //existe
b01110}; 274 proximo menu 303 }
239 269 boolean baixo = ms. 304 void printBatLCD(float lvl) {
240 lcd.createChar(0, batEmpty); 270 printBatLCD(lvl); 305 if (lvl < 6) printBatUSB();
241 lcd.createChar(1, extra); 271 if (cima && baixo){ 306 else if ( lvl < 7.8 )
242 lcd.createChar(2, bat50); 272 lcd.setCursor(15, 1); 307 printBatLow();
243 lcd.createChar(3, bat100); 273 lcd.write(6); 308 else if ( lvl < 9 ) printBat50
244 lcd.createChar(4, setaCima); 274 } 309 ();
245 lcd.createChar(5, setaBaixo); 275 else if (baixo) { 310 else if ( lvl >= 9 )
246 lcd.createChar(6, 276 lcd.setCursor(15, 1); 311 printBat100();
setaCimaBaixo); 277 lcd.write(4); 312 }
247 lcd.createChar(7, ultimo); 278 } 313 void printBatLCD(float lvl) {
248 } 279 else if (cima) { 314 if (lvl < 6) printBatUSB();
249 280 lcd.setCursor(15, 1); 315 else if ( lvl < 7.8 )
250 281 lcd.write(5); 316 printBatLow();
251 /* ----- 282 } 317 else if ( lvl < 9 ) printBat50
LCD - displayMenu - 283 } 318 ();
Mostra sistema de menus no 284 } 319 else if ( lvl >= 9 )
LCD 285 } 320 printBat100();
----- */ 321 void printBatLow(void) {
252 void displayMenu() { 322 lcd.setCursor(15, 0);
253 lcd.setCursor(0,0); 323 lcd.write(0);
254 // Display the menu 286 void battCheckUp(void){ 324 }
255 Menu const* cp_menu = ms. 287 int nivel= analogRead( 325 }
get_current_menu(); 288 BAT_LEVEL_); 326 void battCheckUp(void){
256 289 float tensao0_5= (nivel * 327 int nivel= analogRead(
//lcd.print(" Current menu 290 4.99) / 1024.0; 328 BAT_LEVEL_);
name: "); 291 nivel_de_bat = tensao0_5 * 329 float tensao0_5= (nivel *
258 lcd.print(cp_menu-> 300 11.16 ; // Vout * (R1+R2) 330 4.99) / 1024.0;
get_name()); 301 / R2 = Vin 331 void printBat50(void) {
259 lcd.setCursor(0,1); 302 //desligar caso a bateria 332 lcd.setCursor(15, 0);
260 lcd.write(7); 303 esteja fraca 333 lcd.write(2);
304 }
305 }
306 }
307 }
308 }
309 }
310 }
311 }
312 }
313 }
314 }
315 }
316 }
317 }
318 }
319 }
320 }
321 }
322 }
323 }
324 }

```

325	/* -----	360	else {	398	
	BATERIA – printBat100 –	361	return 0;	399	
	imprime simbolo de bateria	362	}	400	/* -----
	cheia	363			MEDICAO – calibrarBMP
	----- */	364			– fazer zero ao barometro
326	void printBat100(void) {	365			----- */
327	lcd.setCursor(15, 0);	366	/* -----	401	double calibrarBMP(void){
328	lcd.write(3);		SD – saveData – guardar	402	double pressau;
329	}		todos os dados no cartao SD	403	double xHatMinusBMP,
330			----- */		xHatBMP=101325,
331		367	void saveData (double accX,		PminusBMP= 0, PBMP =
332	/* -----		double accY, double pos,		1000, KBMP;
	BATERIA – printBatUSB –		double velo, double acc){	404	double QBMP = 1e-3, RBMP
	imprime simbolo que esta a	368	File ficheiro = SD.open("		= 0.5;
	ser alimentado por USB		Perfil.txt", FILE_WRITE);	405	
	----- */	369	ficheiro . print(millis());	406	xHatBMP = (double) bmp.
333	void printBatUSB(void) {	370	ficheiro . print("\t");		readPressure();
334	lcd.setCursor(15, 0);	371	ficheiro . print(accX,4);	407	
335	lcd.write(1);	372	ficheiro . print("\t");	408	for (int i = 0; i<200; i++){
336	}	373	ficheiro . print(accY,4);	409	xHatMinusBMP=xHatBMP;
337		374	ficheiro . print("\t");	410	PminusBMP= PBMP+QBMP
338		375	ficheiro . print(acc,4);		;
339	/* -----	376	ficheiro . print("\t");	411	KBMP = PminusBMP/(
	TECLADO –	377	ficheiro . print(velo,2);		PminusBMP+RBMP);
	handleKeypad – Intrepreta	378	ficheiro . print("\t");	412	xHatBMP= xHatMinusBMP
	o botao carregado	379	ficheiro . println(pos,2);		+ KBMP * ((double) bmp.
	----- */	380	ficheiro . close();		readPressure() –
340	int handleKeypad(int pin){	381	delay(2);		xHatMinusBMP);
341	// 1 – left	382	}	413	PBMP = (1 – KBMP)*
342	// 2 – up	383			PminusBMP;
343	// 3 – down	384		414	};
344	// 4 – right	385	/* -----	415	return (double) xHatBMP;
345	// 0 – none		SD – saveDataACC –	416	}
346	int valor = analogRead(pin)		guardar apenas valores de	417	
	;		aceleracao no cartao SD	418	
347			----- */	419	/* -----
348	if (valor > 775 && valor <	386	void saveDataACC (double		MEDICAO – medicoes –
	825){		accX, double accY, double		leitura dos sensores
349	return 1;		accZ){		----- */
350	}	387	File ficheiro = SD.open("	420	void medicoes(double* pos,
351	else if (valor > 585 &&		Aceleracoes.txt",		double * aX, double * aY,
	valor < 620){		FILE_WRITE);		double * aZ, double
352	return 2;	388	ficheiro . print(millis());		pressaoRef, int
353	}	389	ficheiro . print("\t");		NAOutilizarBarometro,
354	else if (valor > 370 &&	390	ficheiro . print(accX,4);		double biasAccX, double
	valor < 450){	391	ficheiro . print("\t");		razaoX, double biasAccY,
355	return 4;	392	ficheiro . print(accY,4);		double razaoY, double
356	}	393	ficheiro . print("\t");		biasAccZ, double razaoZ){
357	else if (valor > 180 &&	394	ficheiro . println(accZ,4);	421	int16_t acX=0, acY=0, acZ=0;
	valor < 220){	395	ficheiro . close();	422	double accX=0, accY=0, accZ
358	return 3;	396	delay(2);		=0;
359	}	397	}	423	

```

424 long time = millis ();
425
426 // medicao: 4xMPU + 2
    xBMP + 4xMPU, a media
    permite fazer "filtragem"
427 // duracao: 70 milissegundos
428
429 double temp = mpu.
    getTemperature();
430 mpu.getAcceleration(&acX, &
    acY, &acZ);
431 accX+=(double) acX;
432 accY+=(double) acY;
433 accZ+=(double)acZ;
434
435 temp += mpu.
    getTemperature();
436 mpu.getAcceleration(&acX, &
    acY, &acZ);
437 accX+=(double) acX;
438 accY+=(double) acY;
439 accZ+=(double)acZ;
440
441 temp += mpu.
    getTemperature();
442 mpu.getAcceleration(&acX, &
    acY, &acZ);
443 accX+=(double) acX;
444 accY+=(double) acY;
445 accZ+=(double)acZ;
446
447 temp += mpu.
    getTemperature();
448 mpu.getAcceleration(&acX, &
    acY, &acZ);
449 accX+=(double) acX;
450 accY+=(double) acY;
451 accZ+=(double)acZ;
452
453 if (NAOutilizarBarometro
    == 0){ // utilizar barometro
454 double pressao = (double)
    bmp.readPressure();
455 double posicao = bmp.
    getAltitude(pressao,
    pressaoRef);
456 pressao = (double) bmp.
    readPressure();
457 * pos = (bmp.getAltitude(
    pressao, pressaoRef) +
    posicao)/2;
458 }

459
460 else *pos = 0; //nao utilizar
    barometro
461
462 temp += mpu.
    getTemperature();
463 mpu.getAcceleration(&acX, &
    acY, &acZ);
464 accX+=(double) acX;
465 accY+=(double) acY;
466 accZ+=(double)acZ;
467
468 temp += mpu.
    getTemperature();
469 mpu.getAcceleration(&acX, &
    acY, &acZ);
470 accX+=(double) acX;
471 accY+=(double) acY;
472 accZ+=(double)acZ;
473
474 temp += mpu.
    getTemperature();
475 mpu.getAcceleration(&acX, &
    acY, &acZ);
476 accX+=(double) acX;
477 accY+=(double) acY;
478 accZ+=(double)acZ;
479
480 temp += mpu.
    getTemperature();
481 mpu.getAcceleration(&acX, &
    acY, &acZ);
482 accX+=(double) acX;
483 accY+=(double) acY;
484 accZ+=(double)acZ;
485
486 double temper =(double)((
    temp/8.0)/340 + 36.53);
487
488 * aX = (double)((((double)
    accX/8)+biasAccX)/razaoX
    -0.048835299-0.001-(-0.00589
    temper + 0.0162-0.02));
489 * aY = (double)((((double)
    accY/8)+biasAccY)/razaoY
    +0.048210769+0.001-(0.0051513
    temper - 0.0692));
490 * aZ = (double)((((double)
    accZ/8)+biasAccZ)/
    razaoZ
    -9.81+0.100982619-(-0.0210*
    temper + 0.3846))
    +0.0249891192;
491
492 // manter a frequencia
    time=millis() -time;
493 if (time < DT*1000)
494 delay(DT*1000-time);
495 }
496
497 /* -----
    MEDICAO - medicoes1 -
    leitura dos sensores apos
    fazer o zero
    ----- */
498
499 void medicoes1( double* pos,
    double * aX, double * aY,
    double * aZ, double
    pressaoRef, int modelo,
    double biasX, double biasY,
    double biasZ, double
    biasAccX, double razaoX,
    double biasAccY, double
    razaoY, double biasAccZ,
    double razaoZ){
500 int16_t acX=0, acY=0, acZ=0;
    double accX=0, accY=0, accZ
    =0;
501
502 long time = millis ();
503
504 double temp = mpu.
    getTemperature();
505 mpu.getAcceleration(&acX, &
    acY, &acZ);
506 accX+=(double) acX;
507 accY+=(double) acY;
508 accZ+=(double)acZ;
509
510 temp += mpu.
    getTemperature();
511 mpu.getAcceleration(&acX, &
    acY, &acZ);
512 accX+=(double) acX;
513 accY+=(double) acY;
514 accZ+=(double)acZ;
515
516 }

```

518	temp += mpu. getTemperature();	552	mpu.getAcceleration(&acX, & acY, &acZ);	579	}
519	mpu.getAcceleration(&acX, & acY, &acZ);	553	accX+=(double) acX;	580	
520	accX+=(double) acX;	554	accY+=(double) acY;	581	if (modelo ==2){ // acc no elevador modelo
521	accY+=(double) acY;	555	accZ+= (double)acZ;	582	if (abs(*aX) < 0.02) *aX = 0;
522	accZ+= (double)acZ;	556		583	if (abs(*aY) < 0.02) *aY = 0;
523		557	temp += mpu. getTemperature();	584	if (abs(*aZ) < 0.2) *aZ = 0;
524	temp += mpu. getTemperature();	558	mpu.getAcceleration(&acX, & acY, &acZ);	585	}
525	mpu.getAcceleration(&acX, & acY, &acZ);	559	accX+=(double) acX;	586	
526	accX+=(double) acX;	560	accY+=(double) acY;	587	//manter a frequencia
527	accY+=(double) acY;	561	accZ+= (double)acZ;	588	if (modelo == 0){
528	accZ+= (double)acZ;	562		589	time=millis() –time;
529		563	double temper =(double)((temp/8.0)/340 + 36.53);	590	if (time < DT*1000)
530	if (modelo == 0){ //utilizar Barmometro	564		591	delay(DT*1000–time);
531	double pressao = (double) bmp.readPressure();	565	* aX = (double)(((((double) accX/8)+biasAccX)/razaoX	592	}
532	double posicao = bmp. getAltitude(pressao, pressaoRef);	566	–0.048835299–0.001–(–0.00584 temper + 0.0162–0.02)– biasX/1.5);	593	else {
533	pressao = (double) bmp. readPressure();	567	* aY = (double)(((((double) accY/8)+biasAccY)/razaoY	594	time=millis() –time;
534	* pos = (bmp.getAltitude(pressao, pressaoRef) + posicao)/2;	568	+0.048210769+0.001–(0.0051599 temper – 0.0692)–biasY /1.5);	595	if (time < 20)
535	}	569	* aZ = (double)(((((double) accZ/8)+biasAccZ)/ razaoZ	596	delay(20–time);
536		570	–9.81+0.100982619–(–0.0210* temper + 0.3846)	597	}
537	else *pos = 0; // nao utilizar barometro	571	+0.0249891192–biasZ/1.5);	598	}
538		572		599	
539	temp += mpu. getTemperature();	573		600	/* ----- KALMAN – resetKalman – Reset das variaveis do filtro de kalman ----- */
540	mpu.getAcceleration(&acX, & acY, &acZ);	574		601	
541	accX+=(double) acX;	575		602	void resetKalman(void){
542	accY+=(double) acY;	576		603	xHat_[0]=0;
543	accZ+= (double)acZ;	577		604	xHat_[1]=0;
544		578		605	xHat_[2]=0;
545	temp += mpu. getTemperature();	579		606	xHat_[3]=0;
546	mpu.getAcceleration(&acX, & acY, &acZ);	580		607	xHatMinus_[0]=0;
547	accX+=(double) acX;	581		608	xHatMinus_[1]=0;
548	accY+=(double) acY;	582		609	xHatMinus_[2]=0;
549	accZ+= (double)acZ;	583		610	xHatMinus_[3]=0;
550		584		611	
551	temp += mpu. getTemperature();	585		612	P_[0]=0;
		586		613	P_[1]=0;
		587		614	P_[2]=0;
		588		615	P_[3]=0;
		589		616	
		590		617	Pminus_[0]=0;
		591		618	Pminus_[1]=0;
		592		619	Pminus_[2]=0;
		593		620	Pminus_[3]=0;
		594		621	
		595		622	

```

623 K_[0]=0;
624 K_[1]=0;
625 K_[2]=0;
626 K_[3]=0;
627
628 xHatMinus[0][0]=0;
629 xHatMinus[1][0]=0;
630 xHatMinus[2][0]=0;
631
632 xHat[0][0]=0;
633 xHat[1][0]=0;
634 xHat[2][0]=0;
635
636 P[0][0]=10;
637 P[0][1]=0;
638 P[0][2]=0;
639 P[1][0]=0;
640 P[1][1]=10;
641 P[1][2]=0;
642 P[2][0]=0;
643 P[2][1]=0;
644 P[2][2]=10;
645
646 Pminus[0][0]=0;
647 Pminus[0][1]=0;
648 Pminus[0][2]=0;
649 Pminus[1][0]=0;
650 Pminus[1][1]=0;
651 Pminus[1][2]=0;
652 Pminus[2][0]=0;
653 Pminus[2][1]=0;
654 Pminus[2][2]=0;
655
656 K[0][0]=0;
657 K[0][1]=0;
658 K[1][0]=0;
659 K[2][1]=0;
660 K[2][0]=0;
661 K[1][1]=0;
662 }
663
664
665 /* -----
KALMAN – filtro – filtro
aceleracao apenas
----- */
666 void filtro (double*
aceleracaoZ, double*
aceleracaoY, double*
aceleracaoX){
667 // project the stat ahead
668 xHatMinus_[0]=xHat_[0]; //
ax
669 xHatMinus_[1]=xHat_[1]; //
ay
670 xHatMinus_[2]=xHat_[2]; //
az
671
672 // project the error
covariance ahead
673 // Pminus= A * P * A^T + Q
674 Pminus_[0]= P_[0]+Q_[0]; //
ax
675 Pminus_[1]= P_[1]+Q_[1]; //
ay
676 Pminus_[2]= P_[2]+Q_[2]; //
az
677
678 //Compute kalman gain
679 // K = Pminus * C^T * (C *
Pminus * C^T + R )^-1
680 K_[0] = Pminus_[0]/(
Pminus_[0]+R_[0]); //ax
681 K_[1] = Pminus_[1]/(
Pminus_[1]+R_[1]); //ay
682 K_[2] = Pminus_[2]/(
Pminus_[2]+R_[2]); //az
683
684 //Update estimate with
measurement
685 // xHat = xHatMinus + K ( z
- C * xHatMinus)
686 xHat_[0] = xHatMinus_[0] +
K_[0] * (*aceleracaoX * 1.3
- xHatMinus_[0]); //ax
687 xHat_[1] = xHatMinus_[1] +
K_[1] * (*aceleracaoY * 1.3
- xHatMinus_[1]); //ay
688 xHat_[2] = xHatMinus_[2] +
K_[2] * (*aceleracaoZ * 1.3
- xHatMinus_[2]); //az
689
690 *aceleracaoZ = xHat_[2];
691 *aceleracaoY = xHat_[1];
692 *aceleracaoX = xHat_[0];
693
694 //update error covariance
695 //P = (I - K * C ) * Pminus
696 P_[0] = (1-K_[0])*Pminus_
[0]; //ax
697 P_[1] = (1-K_[1])*Pminus_
[1]; //ay
698 P_[2] = (1-K_[2])*Pminus_
[2]; //az
699 }
700
701 /* -----
KALMAN – filtro2 – filtro
para navegacao inercial
----- */
702 void filtro2 (double* posicao,
double* velocidade, double*
aceleracaoZ, double*
aceleracaoY, double*
aceleracaoX){
703 // project the stat ahead
704 xHatMinus[0][0]=xHat[0][0]+
xHat[1][0]*DT+xHat[2][0]*
DT*DT/2; //pz
705 xHatMinus[1][0]=xHat[1][0]+
xHat[2][0]*DT; //vz
706 xHatMinus[2][0]=xHat[2][0];
//az
707
708 xHatMinus_[0]=xHat_[0]; //
ax
709 xHatMinus_[1]=xHat_[1]; //
ay
710 xHatMinus_[3]=xHat_[3]; //
pz
711
712 // project the error
covariance ahead
713 // Pminus= A * P * A^T + Q
714 Pminus[0][0]= constrain(( P
[2][2]* DT*DT*DT*DT+2*P
[1][2]*DT*DT*DT+2*P
[2][1]*DT*DT*DT+2*P
[0][2]*DT*DT+4*P[1][1]*DT
*DT+2*P[2][0]*DT*DT+4*P
[0][1]*DT+4*P[0][0])/4 +
WITHE_NOISE_PSD * DT*
DT*DT*DT/20, -1.0e9,
1.0e9);
715 Pminus[1][0]= constrain(2*( P
[2][2]* DT*DT*DT+P[1][2]*
DT*DT+2*P[2][1]*DT*DT
+2*P[1][1]*DT+2*P[2][0]*
DT+2*P[1][0])/4 +
WITHE_NOISE_PSD * DT*
DT*DT*DT/8, -1.0e9, 1.0
e9);
716
717

```


718	Pminus[2][0]= constrain (2*(P	736	K[1][0]= (Pminus[1][2]/(766	
	[2][2]* DT*DT+2*P[2][1]*		Pminus[2][2]+0.004));	767	//update error covariance
	DT+2*P[2][0])/4 +	737	K[2][0]= (Pminus[2][2]/(768	//P = (I - K * C) * Pminus
	WITHE_NOISE_PSD * DT*		Pminus[2][2]+0.004));	769	P[0][0]= Pminus[0][0]-
	DT*DT/6, -1.0e9, 1.0e9);	738			Pminus[2][0]*K[0][0];
719		739	K_[0] = Pminus_[0]/(770	P[1][0]= Pminus[1][0]-
720	Pminus[0][1]= constrain ((P		Pminus_[0]+R_[0]); //ax		Pminus[2][0]*K[1][0];
	[2][2]* DT*DT*DT+2*P	740	K_[1] = Pminus_[1]/(771	P[2][0]= Pminus[2][0]*(1-K
	[1][2]*DT*DT+P[2][1]*DT*		Pminus_[1]+R_[1]); //ay		[2][0]);
	DT+2*P[0][2]*DT+2*P	741	K_[3] = Pminus_[3]/(772	
	[1][1]*DT+2*P[0][1])/2+		Pminus_[3]+R_[3]); //pz	773	P[0][1]= Pminus[0][1]-
	WITHE_NOISE_PSD * DT*	742			Pminus[2][1]*K[0][0];
	DT*DT*DT/8, -1.0e9, 1.0	743	float factor=1.3;	774	P[1][1]= Pminus[1][1]-
	e9);	744			Pminus[2][1]*K[1][0];
721	Pminus[1][1]= constrain (P	745	//Update estimate with	775	P[2][1]= Pminus[2][1]*(1-K
	[2][2]* DT*DT+P[1][2]*DT+		measurement		[2][0]);
	P[2][1]*DT+P[1][1]+	746	// xHat = xHatMinus + K (z	776	
	WITHE_NOISE_PSD * DT*		- C * xHatMinus)	777	P[0][2]= Pminus[0][2]-
	DT*DT/3, -1.0e9, 1.0e9);	747	xHat[0][0]=xHatMinus[0][0]+		Pminus[2][2]*K[0][0];
722	Pminus[2][1]= constrain (P		K[0][0]*(*aceleracaoZ*	778	P[1][2]= Pminus[1][2]-
	[2][1]+P[2][2]* DT+		factor-xHatMinus[2][0]);		Pminus[2][2]*K[1][0];
	WITHE_NOISE_PSD * DT*	748	xHat[1][0]=xHatMinus[1][0]+	779	P[2][2]= Pminus[2][2]*(1-K
	DT/2, -1.0e9, 1.0e9);		K[1][0]*(*aceleracaoZ*		[2][0]);
723			factor-xHatMinus[2][0]);	780	
724	Pminus[0][2]= constrain ((P	749	xHat[2][0]=xHatMinus[2][0]+	781	P_[0] = (1-K_[0])*Pminus_
	[2][2]* DT*DT/2+P[1][2]*		K[2][0]*(*aceleracaoZ*		[0]; //ax
	DT+2*P[0][2])+		factor-xHatMinus[2][0]);	782	P_[1] = (1-K_[1])*Pminus_
	WITHE_NOISE_PSD * DT*	750			[1]; //ay
	DT*DT/6, -1.0e9, 1.0e9);	751	xHat_[0] = xHatMinus_[0] +	783	P_[3] = (1-K_[3])*Pminus_
725	Pminus[1][2]= constrain (P		K_[0] * (*aceleracaoX -		[3]; //pz
	[1][2]+P[2][2]* DT+	752	xHatMinus_[0]); //ax	784	}
	WITHE_NOISE_PSD * DT*		xHat_[1] = xHatMinus_[1] +	785	
	DT/2, -1.0e9, 1.0e9);		K_[1] * (*aceleracaoY -	786	
726	Pminus[2][2]= constrain (P		xHatMinus_[1]); //ay	787	/* -----
	[2][2]+	753	xHat_[3] = xHatMinus_[3] +		MENUS -
	WITHE_NOISE_PSD * DT,		K_[3] * (*posicao * 1.05 -		on_item1_selected -
	-1.0e9, 1.0e9);	754	xHatMinus_[3]); //pz		determinar perfis
727					----- */
728	Pminus_[0]= P_[0]+Q_[0]; //	755		788	void on_item1_selected(
	ax	756	float polo = 0.95;		MenuItem* p_menu_item){
729	Pminus_[1]= P_[1]+Q_[1]; //	757	//compensar erros de	789	lcd. clear ();
	ay		integracao	790	printBatLCD(nivel_de_bat);
730	Pminus_[3]= P_[3]+Q_[3]; //	758	xHat[0][0] = xHat[0][0] *	791	lcd. print ("Fazer zero... ");
	pz		(1-polo) + (xHat_[3] *	792	pressaoReferencia = (double)
731			polo) ;		calibrarBMP());
732		759		793	
733	//Compute kalman gain	760	*posicao= xHat[0][0];	794	resetKalman();
734	// K = Pminus * C^T * (C *	761	*velocidade = xHat[1][0];	795	
	Pminus * C^T + R)^-1	762	*aceleracaoZ = xHat[2][0];	796	int j=0;
735	K[0][0]= (Pminus[0][2]/(763	*aceleracaoY = xHat_[1];	797	double BIASx = 0, BIASy = 0,
	Pminus[2][2]+0.004));	764	*aceleracaoX = xHat_[0];		BIASz = 0;
		765		798	for (int i=0; i<50; i++) {

```

799 medicoes( &pos, &      830 lcd.print(pos,2);      869 myFile.println("\n\n acX \t
    aceleracaoX, &      831 lcd.print(" V:");
    aceleracaoY, &      832 lcd.print(velocidade,2);      870 myFile.close();
    aceleracaoZ, 0,1);      833 lcd.newLine();      871
800 filtro (&aceleracaoZ, &      834 lcd.print("accZ:");      872 int notaRMS;
    aceleracaoY, &      835 lcd.print(aceleracaoZ,2);      873 int i=0;
    aceleracaoX);      836 }      874 double rms = 0;
801 BIASx +=aceleracaoX;      837 }      875
802 BIASy +=aceleracaoY;      838      876 double rmq = 0;
803 BIASz +=aceleracaoZ;      839      877 int j=0;
804 j++;      840 /* -----      878 int notaRMQ;
805 }      841 MENUS -      879
806      842 on_item2_selected -      880 lcd.clear();
807 BIASx = BIASx/49;      843 avaliacao NDE      881 printBatLCD(nivel_de_bat);
808 BIASy = BIASy/49;      844 ----- */      882 lcd.print("Avaliando...");
809 BIASz = BIASz/49;      841 void on_item2_selected(      883
810      842 MenuItem* p_menu_item){      884 int tempInicio=millis();
811 lcd.clear();      843 lcd.clear();      885 int duracao = 0;
812 printBatLCD(nivel_de_bat);      844 printBatLCD(nivel_de_bat);      886 int time = 0;
813 lcd.print("Zero feito!");      845 lcd.print("Fazer zero... ");      887 int contador = 0;
814 resetKalman();      846      888
815      847 resetKalman();      889 resetKalman();
816 delay(500);      848      890
817 myFile = SD.open("Perfil.txt",      849 double BIASx = 0, BIASy = 0,      891 while (handleKeypad(
    FILE_WRITE);      850 BIASz = 0;      892 KEYPAD_)==0){
818 myFile.println("\n\n time \n      851 for(int i=0; i<50; i++) {      893 time = millis();
    accX \t accY \t pos \t vel      852 medicoes( &pos, &      894 medicoes1( &pos, &
    \t accZ");      853 aceleracaoX, &      895 aceleracaoX, &
819 myFile.close();      854 aceleracaoY, &      896 aceleracaoY, &
820      855 aceleracaoZ, 0,1);      897 aceleracaoZ, 0,1,BIASx,
821 //medir dados ate ser      856 filtro (&aceleracaoZ, &      898 BIASy,BIASz);
    carregado no keyPad      857 aceleracaoY, &      899
822 while (handleKeypad(      858 aceleracaoX);      900 duracao = millis() - time;
    KEYPAD_)==0){      859 BIASx +=aceleracaoX;      901 filtro ( &aceleracaoZ, &
823 medicoes1( &pos, &      860 BIASy +=aceleracaoY;      902 aceleracaoY, &
    aceleracaoX, &      861 BIASz +=aceleracaoZ;      903 aceleracaoX);
    aceleracaoY, &      862 }      904
    aceleracaoZ,      863 delay(200);      905
    pressaoReferencia,0,BIASx      864 BIASx = BIASx/49;
    ,BIASy,BIASz);      865 BIASy = BIASy/49;
824 filtro2 (&pos, &velocidade,      866 BIASz = BIASz/49;      906
    &aceleracaoZ, &      867 lcd.clear();
    aceleracaoY, &      868 printBatLCD(nivel_de_bat);
    aceleracaoX);      869 lcd.print("Zero feito!");
825 saveData (aceleracaoX,      870 delay(500);
    aceleracaoY, pos,      871
    velocidade, aceleracaoZ      872
    );      873
826      874 myFile = SD.open("
827 lcd.clear();      875 Aceleracoes.txt",
828 printBatLCD(nivel_de_bat);      876 FILE_WRITE);
829 lcd.print("P:");      877 myFile.println("");

```


900	rmq+= (acceleracaoX * acceleracaoX * acceleracaoX * acceleracaoX + acceleracaoY * acceleracaoY * acceleracaoY * acceleracaoY + acceleracaoZ * acceleracaoZ * acceleracaoZ * acceleracaoZ);	928	myFile = SD.open("Aceleracoes.txt", FILE_WRITE);	968	/* ----- MENUS -
901	contador ++;	929	myFile.print("***** \t NDE (RMS): ");	969	on_item4_selected - desligar auto-alimentacao ----- */
902		930	myFile.println(notarMS);	970	void on_item4_selected(MenuItem* p_menu_item){
903	lcd.clear();	931	myFile.print("NDE (RMQ): ");	971	lcd.clear();
904	printBatLCD(nivel_de_bat);	932	myFile.println(notarMQ);	972	lcd.print("Adeus");
905	lcd.print("Avaliando...");	933	myFile.close();	973	delay(500);
906	lcd.newLine();	934	delay(2);	974	//desliagr Auto alimentacao
907	lcd.print("acc Z: ");	935		975	digitalWrite(POWER_SUPPLY_, LOW);
908	lcd.print(acceleracaoZ,2);	936	lcd.clear();	976	}
909	}	937	printBatLCD(nivel_de_bat);	977	
910		938	lcd.print("NDE (rms): ");	978	/* ----- MENUS -
911	duracao = millis() - tempInicio;	939	lcd.print(notarMS);	979	on_item5_selected - tensao na bateria ----- */
912		940	lcd.newLine();	980	void on_item5_selected(MenuItem* p_menu_item){
913	float RMS = (float) sqrt(rms / contador);	941	lcd.print("NDE (rmq): ");	981	float percent;
914	float RMQ = (float) pow((rmq / contador),0.25);	942	lcd.print(notarMQ);	982	int nivel = 0;
915		943	delay(1500);	983	delay(300);
916	if (RMS < 0.3) notarMS = 5;	944	while (handleKeypad(KEYPAD_)==0){	984	
917	else if (RMS < 0.6)	945	}	985	while (handleKeypad(KEYPAD_)==0){
918	notarMS = 4;	946	}	986	nivel= analogRead(BAT_LEVEL_);
919	else if (RMS < 1) notarMS = 3;	947		987	
920	else if (RMS < 1.3)	948	/* ----- MENUS -	988	float tensao0_5= (nivel * 4.99) / 1024.0;
921	notarMS = 2;	949	on_item3_selected - ligar/ desligar luz do LCD ----- */	989	float Vin = tensao0_5 * 11.16 ; // Vout * (R1+R2) / R2 = Vin
922	else notarMS = 1;	950	void on_item3_selected(MenuItem* p_menu_item){	990	
923	if (RMQ < 0.3) notarRMQ = 5;	951	int dentro = 0;	991	battery.handleRun();
924	else if (RMQ < 0.6)	952	//desligar LUZ	992	
925	notarRMQ = 4;	953	if (backLight == 0 && dentro == 0) {	993	lcd.clear();
926	else if (RMQ < 1.)	954	lcd.noBacklight();	994	printBatLCD(nivel_de_bat);
927	notarRMQ = 3;	955	backLight = 1;	995	lcd.print("Tensao: ");
	else if (RMQ < 1.3)	956	dentro=1;	996	lcd.print(Vin, 1);
	notarRMQ = 2;	957	}	997	delay(200);
	else notarRMQ = 1;	958		998	}
		959	//ligar LUZ	999	}
		960	if (backLight == 1 && dentro == 0) {	1000	
		961	lcd.backlight();	1001	
		962	backLight = 0;		
		963	dentro=1;		
		964	}		
		965	delay(100);		
		966	}		
		967			

1002	/* -----	1024	delay(500);	1037	if (valorPWM <0)
	MENUS –	1025			valorPWM = 0;
	on_item6_selected –	1026	double somatorio=0.0;	1038	if (valorPWM >255)
	aceleracao PWM	1027	int valorPWM = 0;		valorPWM = 255;
	----- */	1028	int contador = 0;	1039	
1003	void on_item6_selected(1029		1040	lcd.clear();
	MenuItem* p_menu_item){	1030	while (handleKeypad(1041	printBatLCD(nivel_de_bat);
1004			KEYPAD_)==0){	1042	lcd.print("PWM: ");
1005	printBatLCD(nivel_de_bat);	1031	medicoes1(&pos, &	1043	lcd.print(valorPWM);
1006	lcd.print("Fazer zero... ");		aceleracaoX, &	1044	lcd.newLine();
1007			aceleracaoY, &	1045	lcd.print("Acc: ");
1008	resetKalman();		aceleracaoZ, 0,2,BIASx,	1046	lcd.print(aceleracaoZ,2);
1009			BIASy,BIASz);	1047	analogWrite(PWM_PORTA_,
1010	double BIASx = 0, BIASy = 0,	1032	filtro (&aceleracaoZ, &		valorPWM); //
	BIASz = 0;		aceleracaoY, &		analogRead values go
1011	for(int i=0; i<100; i++) {		aceleracaoX);		from 0 to 1023,
1012	medicoes(&pos, &	1033	somatorio+= (aceleracaoX		analogWrite values from 0
	aceleracaoX, &		* aceleracaoX *		to 255
	aceleracaoY, &		aceleracaoX *	1048	}
	aceleracaoZ, 0,1);		aceleracaoX +	1049	
1013	filtro (&aceleracaoZ, &		aceleracaoY *	1050	analogWrite(PWM_PORTA_,
	aceleracaoY, &		aceleracaoY *		0);
	aceleracaoX);		aceleracaoY *	1051	lcd.clear();
1014	BIASz +=aceleracaoZ;		aceleracaoY +	1052	printBatLCD(nivel_de_bat);
1015	}		aceleracaoZ *	1053	float valorRMQ = (float)
1016	delay(200);		aceleracaoZ *		pow((somatorio / contador
1017	BIASz = BIASz/99;		aceleracaoZ *),0.25);
1018			aceleracaoZ);	1054	lcd.print("NDE (rmq: ");
1019	resetKalman();	1034	contador ++;	1055	lcd.print(valorRMQ,2);
1020		1035		1056	delay(2000);
1021	lcd.clear();	1036	valorPWM = (int) round(1057	}
1022	printBatLCD(nivel_de_bat);		255 * (aceleracaoZ +1.2)		
1023	lcd.print("Zero feito!");		/2.4);		